THE UNIVERSITY OF MICHIGAN

COLLEGE OF ENGINEERING

Department of Aeronautical and Astronautical Engineering

Technical Report

THE USE OF STATISTICAL DEPENDENCE BETWEEN DATA SAMPLES IN BINARY PCM DEMODULATION

James Kinard Strozier

NASA Research Grant NsG-2-59

ORA Project 02905

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

November 1966

THE USE OF STATISTICAL DEPENDENCE BETWEEN DATA SAMPLES IN BINARY PCM DEMODULATION

by
James Kinard Strozier

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in the
University of Michigan
1966

Doctoral Committee:

Professor Lawrence L. Rauch, Chairman
Professor Elmer G. Gilbert
Professor Robert M. Howe
Professor William A. Porter
Professor William L. Root

Acknowledgements

A dissertation of this nature would not be possible without the assistance and encouragement of many other people. The author is particularly indebted to the members of his doctoral committee, Professors L. L. Rauch, E. G. Gilbert, R. M. Howe, W. A. Porter and W. L. Root, whose patience and helpful criticisms were invaluable. I am especially grateful to Professor L. L. Rauch, who as chairman of the committee gave me so much encouragement and assistance that this work would be nonexistent without his help.

The author would also like to express his gratitude to the United States Army and the United States Military Academy for making this dissertation possible. In particular, I would like to thank Colonel E. R. Heiberg, the Head of the Department of Mechanics, U.S.M.A., for making the time available for me to complete this dissertation and for his encouragement to do so. Special thanks are due Brig. Gen. H. R. Fraser (Ret.), whose dynamic encouragement kept me going when the progress was slowest, and the staff of the U.S.M.A. Academic Computer Center, under the direction of Lt. Col. W. F. Luebbert, whose advice and assistance in the programming were invaluable.

Last but certainly not least, the author would like to thank the University of Michigan Computing Center for their assistance in utilizing their electronic computing facilities, Miss Linda Swenson for her patience and help in preparing the draft manuscript, and Mr. John Parsons for the excellent preparation of the final manuscript.

Table of Contents

INTRODUCTION

Due to the widespread use of pulse code modulation (PCM) for data transmission, there is considerable interest in the improvement of the demodulation process. Most present day demodulators demodulate a received PCM signal by considering one bit of each word at a time, and therefore make no use of the statistical dependence between data samples, although for reasonable interpolation errors, the required sampling rates are sufficiently high to give a high correlation between samples in many applications. For example, it is shown in Appendix I that when 6-bit words are used to transmit second order Butterworth data, the required sampling rates are such that the correlation coefficient between adjacent samples is greater than .98 and the correlation coefficient between every second sample is greater than .95. One would expect that such high correlation between data samples could be used to improve the demodulation process. Smith [S1, S2][1] has shown that this high correlation between samples can be used to improve the word-error performance of the demodulator that minimizes the probability of an error in the demodulation process.

This dissertation will treat the use of statistical dependence between PCM data samples in several optimal and suboptimal demodulators. Three optimal demodulators will be investigated: the minimum error-probability (P_e) demodulator, the minimum mean-absolute-error (MAE) demodulator, and the minimum mean-square-error (MSE) demodulator. Three suboptimal demodulation schemes will also be investigated.

The following restrictions and assumptions, which are essentially those used by Smith [S1, S2], will be adhered to in this analysis. Only single data sources will be considered, but the extension to multiple

[1] Letter-number combinations in brackets refer to references listed in the bibliography.

data sources is obvious. The analog data samples will be assumed to come from a Gaussian random process with known autocorrelation. The assumption of Gaussian data is immaterial in the general development of the demodulators (Chapter 1), and becomes important only when the simulation on the computer is begun (Chapter 3). As will be apparent in Chapter 3, data samples from any random process could be considered, provided there is a way to generate correlated samples from the distribution on the computer and some way to evaluate the multivariate probability for a given set of quantized data samples. The channel noise will be considered additive only, and independent of the transmitted signal. The noise is further assumed to be bandlimited, white, Gaussian noise. Although this last assumption was made so that analytic progress is possible, in practice the noise power spectrum is essentially flat out to some arbitrarily high frequency and is bandlimited by the equipment. Therefore, the assumption of bandlimited white noise is a reasonable one to make.

In evaluating the performance of the demodulators, the demodulated signal will be compared with the quantized data sample that was transmitted, rather than with the original analog data sample. This is reasonable since quantization errors before transmission should not be charged to the demodulator.

CHAPTER 1

OPTIMAL DEMODULATION OF BINARY PCM WAVEFORMS

The notation used is that indicated in figure 1-1. $x(t)$ is the analog data signal that will be transmitted using binary PCM signals. As mentioned in the introduction, $x(t)$ is assumed to be a stationary Gaussian random process with specified autocorrelation function from which are obtained the correlation coefficients ρ_{ij} between samples x_i^* and x_j^*. The samples $x^*(t)$ of $x(t)$ are quantized into discrete value data, $Y(t)$. This discrete-value data is then coded into serial binary PCM signals, $y(t)$. Gaussian white noise, $n(t)$, is added in the transmission process, so that the received signal $z(t)$ is the sum of $y(t)$ and $n(t)$. $z(t)$ is then demodulated to get $\hat{Y}(t)$, the estimate of $Y(t)$. The remainder of the process to get $\hat{x}(t)$, will not be considered as part of the demodulation process in this dissertation, although it certainly plays an important role in the overall data transmission system. The next-to-last box in the block diagram is not labeled since in some cases (e.g., the minimum MSE demodulator) the digital-to-analog conversion is accomplished in the demodulator, and in most actual systems the sampled signal $\hat{x}^*(t)$ is taken equal to $\hat{Y}(t)$.

The binary PCM signal, $y(t)$, will be considered to be a series of equal energy waveforms $g_1(t)$ or $g_2(t)$, $g_1(t)$ representing a "yes" bit, and $g_2(t)$ a "no" bit. This is easily generalized to other bit waveforms (See Appendix II). Let T_W be the time required to transmit one PCM word, and let m be the number of bits in each word. Then each "word-time", T_W, is divided into m equal parts called "bit-times" and denoted by T_B.

In an m-bit binary code there are 2^m quantization levels, or 2^m possible values of Y. For simplicity we take these values of Y to be

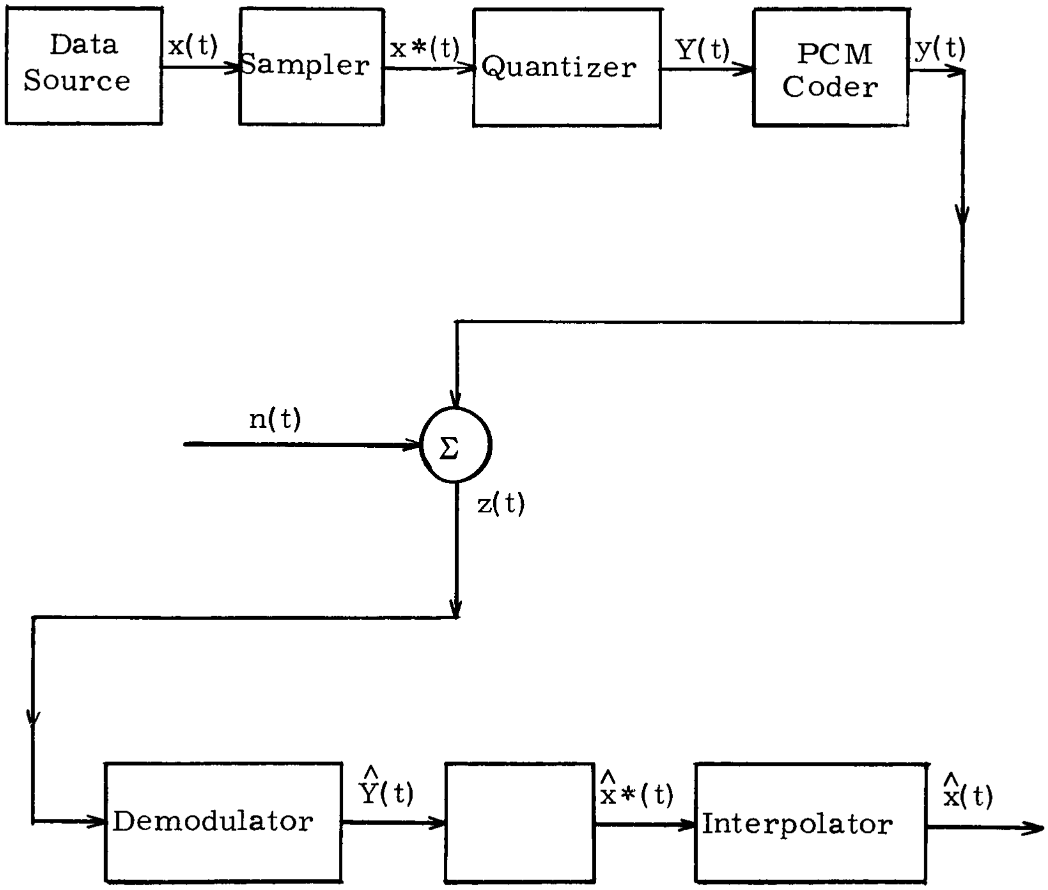

Figure 1-1. System Block Diagram

the integers $k = 0, 1, \ldots, 2^m - 1$ and scale $x(t)$ accordingly. Let the joint probability of the quantized value, Y_j, of the jth sample and the received signal $z(t)$ be given by

$$f_{Y_j z}(k, z(t)) = f_{Y_j|z}(k|z(t))\, f_z(z(t))$$
$$= f_{z|Y_j}(z(t)|k) f_{Y_j}(k)$$

where the probability is discrete in k and a multivariant density in a time sampled representation of $z(t)$.

Let us now consider the three types of optimal demodulators that will be analyzed in this dissertation. The minimum error-probability demodulator minimizes the average probability of error, P_e. If the demodulation operator is given by $\hat{Y}_j[z(t)]$, then the error-probability for a particular $z(t)$ is $1 - f_{Y_j|z}(\hat{Y}_j[z(t)]\,|\,z(t))$ and

$$P_e = \int_z (1 - f_{Y_j|z}(\hat{Y}_j[z(t)]\,|\,z(t))) f_z(z(t))\, dz$$
$$= 1 - \int_z f_{Y_j|z}(\hat{Y}_j[z(t)]\,|\,z(t)) f_z(z(t))\, dz.$$

Clearly, this is minimized by choosing $\hat{Y}_j[z(t)]$ to maximize $f_{Y_j|z}(\hat{Y}_j[z(t)]\,|\,z(t))$ for every $z(t)$. Thus the minimum error-probability demodulator is described by

$$\hat{Y}_{j1}[z(t)] = k \text{ where } k \text{ maximizes } f_{Y_j|z}(k|z(t)). \qquad (1\text{-}1)$$

This $\hat{Y}_{j1}$ also maximizes the probability of correct demodulation and it is often called the maximum inverse-probability demodulator.

The mean absolute error (MAE) of a demodulator is given by

$$\text{MAE} = E[\,|Y_j - \hat{Y}_j[z(t)]|\,]$$

where E[ ] denotes the expectation operation. In terms of the previous probabilities

$$\mathrm{MAE} = \int_z \sum_{k=0}^{2^m-1} |k - \hat{Y}_j[z(t)]| f_{Y_j z}(k, z(t))\, dz$$

$$= \int_z \sum_{k=0}^{2^m-1} |k - \hat{Y}_j[z(t)]| f_{Y_j|z}(k|z(t)) f_z(z(t))\, dz.$$

Clearly, this is minimized by choosing $\hat{Y}_j[z(t)]$ to minimize

$$\sum_{k=0}^{2^m-1} |k - \hat{Y}_j[z(t)]| f_{Y_j|z}(k|z(t))$$

for every z(t). It is well known [B1] that the required $\hat{Y}_j[z(t)]$ is given by the median of the conditional distribution $f_{Y_j|z}(k|z(t))$ where the median of a real-valued random variable v is defined as follows. Let A be the real line, let

$$m_1 = \inf\{a \in A \,|\, P(v \leqq a) \geqq 1/2\}$$
$$m_2 = \sup\{a \in A \,|\, P(v \geqq a) \geqq 1/2\}$$

and let I be the closed interval $[m_1, m_2]$. Then any $m \in I$ is called a median of the random variable v. The set of z(t) giving non-unique medians has probability zero in this application. Thus the minimum MAE demodulator is described by

$$\hat{Y}_{j2}[z(t)] = \text{median of } f_{Y_j|z}(k|z(t)). \qquad (1\text{-}2)$$

The mean square error (MSE) of a demodulator is given by

$$\mathrm{MSE} = \int_z \sum_{k=0}^{2^m-1} (k - \hat{Y}_j[z(t)])^2 f_{Y_j|z}(k|z(t)) f_z(z(t))\, dz$$

This is minimized by choosing $\hat{Y}_j[z(t)]$ to minimize

$$\sum_{k=0}^{2^m-1} (k - \hat{Y}_j[z(t)])^2 f_{Y_j|z}(k|z(t))$$

for every z(t). This is accomplished by taking $\hat{Y}_j[z(t)]$ equal to the mean of the conditional distribution

$$\hat{Y}_{j3}[z(t)] = \sum_{k=0}^{2^m-1} k\, f_{Y_j|z}(k|z(t)). \qquad (1\text{-}3)$$

Equations 1-1, 1-2, and 1-3 all involve the conditional probability, $f_{Y_j|z}(k|z(t))$. Smith has derived an expression for this conditional probability [S1] and this derivation is repeated here for convenience. In the following, the first subscript refers to the word position in the PCM signal sequence, and the second subscript to the bit position in the PCM word (e.g., y_{ir} is the transmitted signal during the rth bit time of the ith word time).

The conditional probability of the sequence of quantized data samples, $Y_1, \ldots, Y_n$ given $z_1, \ldots, z_n$ is given by

$$f_{Y|z}(Y_1, \ldots, Y_n|z_1, \ldots, z_n) = \frac{f_{Yz}(Y_1, \ldots, Y_n, z_1, \ldots, z_n)}{f_z(z_1, \ldots, z_n)}$$

$$= \frac{f_{z|Y}(z_1, \ldots, z_n|Y_1, \ldots, Y_n) f_Y(Y_1, \ldots, Y_n)}{f_z(z_1, \ldots, z_n)} \qquad (1\text{-}4)$$

Since there is a unique correspondence between Y_i and y_i

$$f_{z|Y}(z_1, \ldots, z_n|Y_1, \ldots, Y_n) = f_{z|y}(z_1, \ldots, z_n|y_1, \ldots, y_n) \qquad (1\text{-}5)$$

The channel noise, n, was assumed additive and independent of the transmitted signal so that

$$f_{z|y}(z_1, \ldots, z_n|y_1, \ldots, y_n) = f_n(z_1 - y_1, \ldots, z_n - y_n) \qquad (1\text{-}6)$$

With proper sampling, bandlimited white Gaussian noise is independent from sample to sample. Therefore,

$$f_n(z_1 - y_1, \ldots, z_n - y_n) = f_n(z_1 - y_1) \cdots f_n(z_n - y_n) \tag{1-7}$$

Woodward has shown [W1] that

$$f_n(z_i - y_i) = \frac{1}{\sqrt{2\pi}\,\sigma_n} \exp\left\{ \frac{-1}{2\sigma_n^2} \int_{T_W} (z_i - y_i)^2\, dt \right\} \tag{1-8}$$

Expanding the integral in equation 1-8 and noting that $\int_{T_W} y_i^2\, dt$ is a constant for the equal energy signals that we are considering and $\int_{T_W} z_i^2\, dt$ is a function only of z_i, we have

$$f_n(z_i - y_i) = K_1(z_i) \exp\left\{ \frac{1}{\sigma_n^2} \int_{T_W} z_i y_i\, dt \right\} \tag{1-9}$$

where $K_1(z_i)$ is some function of z_i.

The integral in equation 1-9 can be expanded as

$$\int_{T_W} z_i y_i\, dt = \sum_{r=1}^{m} \int_{T_B} z_{ir} y_{ir}\, dt \tag{1-10}$$

Using equations 1-6 through 1-10, equation 1-5 can be written as

$$f_{z|Y}(z_1, \ldots, z_n | Y_1, \ldots, Y_n) = K_2(z) \prod_{i=1}^{n} \exp\left\{ \frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{ir} y_{ir}\, dt \right\} \tag{1-11}$$

where $K_2(z)$ is a function of the received signals, $z_1, \ldots, z_n$.

The desired conditional probability is given by

$$f_{Y_j|z}(k|z(t)) = \sum_{Y_1 \epsilon U} \cdots \sum_{Y_{j-1} \epsilon U} \sum_{Y_{j+1} \epsilon U} \cdots \sum_{Y_n \epsilon U} f_{Y|z}(Y_1, \ldots, Y_{j-1},$$

$$k, Y_{j+1}, \ldots, Y_n | z_1, \ldots, z_n) \tag{1-12}$$

where U is the set of all possible values $(0, 1, \ldots, 2^m - 1)$ of Y_i.

Denoting by $y_{i(k)r}$ the rth bit of the PCM code corresponding to $Y_i = k$ and noting that $f_z(z_1, \ldots, z_n)$ is only a function of z, we can combine equations 1-12, 1-11, and 1-4 to get

$$f_{Y_j|z}(k|z(t)) = \sum_{Y_1 \epsilon U} \cdots \sum_{Y_{j-1} \epsilon U} \sum_{Y_{j+1} \epsilon U} \cdots \sum_{Y_n \epsilon U} f_Y(Y_1, \ldots, Y_{j-1},$$

$$k, Y_{j+1}, \ldots, Y_n) \cdot K_3(z) \prod_{\substack{i=1 \\ i \neq j}}^{n} \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{ir} y_{ir}\, dt\right\} \exp$$

$$\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{jr} y_{j(k)r}\, dt\right\}$$

$$= K_3(z) \sum_{\ell_1=0}^{2^m-1} \cdots \sum_{\ell_{j-1}=0}^{2^m-1} \sum_{\ell_{j+1}=0}^{2^m-1} \cdots \sum_{\ell_n=0}^{2^m-1} f_Y(\ell_1, \ldots, \ell_{j-1}, k, \ell_{j+1}, \ldots, \ell_n)$$

$$\cdot \prod_{\substack{i=1 \\ i \neq j}}^{n} \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{ir} y_{i(\ell_i)r}\, dt\right\} \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{jr} y_{j(k)r}\, dt\right\} \tag{1-13}$$

Equations 1-1, 1-2, 1-3, and 1-13 completely specify the three optimal demodulators. First the 2^m values of $f_{Y_j|z}$ would be calculated from equation 1-13. The minimum PE demodulator would simply choose the value of Y_j that corresponds to the largest value of the conditional probability, $f_{Y_j|z}$. The minimum MAE demodulator would sum in order the values of $f_{Y_j|z}$, taking $\hat{Y}_{j2}$ to be the value of Y_j that corresponds to the value of the conditional probability that makes the sum equal to or just go past .5. The minimum MSE demodulator would perform the operation indicated by equation 1-3 to get $\hat{Y}_{j3}$. It should be noted that the minimum PE and MAE demodulators can only select $\hat{Y}_j$ to be one of the possible values of Y_j, while this is not true for the minimum MSE demodulator.

CHAPTER 2

PERFORMANCE EVALUATION FOR THE OPTIMAL DEMODULATORS

In Chapter 1 the equations describing the three optimal demodulators that we are considering were developed. To evaluate the performance of these demodulators, it is necessary to have some criteria of goodness with which to measure their performance. The two most widely used criteria are the MAE and the MSE (the root-mean-square-error (RMSE) is often used in place of the MSE), and these two measures of demodulator performance will be used in this analysis.

The error in the demodulation of the PCM signal during the jth word time, e_j, is given by

$$e_j = Y_j - \hat{Y}_j \tag{2-1}$$

and

$$MAE = E\{|e_j|\} = E\{|Y_j - \hat{Y}_j|\} \tag{2-2a}$$

$$MSE = E\{e_j^2\} = E\{(Y_j - \hat{Y}_j)^2\} \tag{2-2b}$$

where, as before, E represents the expectation operator. For simplicity, we will consider only the MAE in what follows, noting that there is a parallel development for the MSE. Equation 2-2a can be written as

$$MAE = \sum_{Y_j \in U} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} |Y_j - \hat{Y}_j| f_{Y_j Z}(Y_j, z_1, \ldots, z_n)\, dz_1, \ldots, dz_n$$

$$= \sum_{Y_1 \in U} \cdots \sum_{Y_n \in U} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} |Y_j - \hat{Y}_j| f_{YZ}(Y_1, \ldots, Y_n, z_1, \ldots, z_n) dz_1, \ldots, dz_n$$

$$= \sum_{Y_1 \epsilon U} \cdots \sum_{Y_n \epsilon U} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} |Y_j - \hat{Y}_j| f_{z|Y}(z_1, \ldots, z_n | Y_1, \ldots, Y_n)$$

$$\cdot dz_1, \ldots, dz_n f_Y(Y_1, \ldots, Y_n)$$

Since $z_i = y_i + n_i$, and there is a one-to-one correspondence between Y_i and y_i, integration over the z space is equivalent to integration over the n space. Using the independence of y and n,

$$\text{MAE} = \sum_{Y_1 \epsilon U} \cdots \sum_{Y_n \epsilon U} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} [\,|Y_j - \hat{Y}_j| f_n(n_1, \ldots, n_n)]\, dn_1, \ldots, dn_n$$

$$\cdot f_Y(Y_1, \ldots, Y_n) \qquad (2\text{-}3)$$

where $\hat{Y}_j$ is a function of $Y_1, \ldots, Y_n$ and $n_1, \ldots, n_n$. Since n_i can be broken down into m components, this expression represents 2^{mn} summations of an $n \cdot m$ dimensional integral. Attempts to analytically evaluate equation 2-3 prove fruitless, even in the most analytically tractable case of the minimum MSE demodulator.

Numerical integration of equation 2-3 could be used. If k sub-intervals of each n_{ir} were used in the integration, then this equation would represent at least k^{nm} evaluations of the quantity in the brackets, each of which includes a determination of $\hat{Y}_j$ by equations 1-13 and 1-1, 1-2, or 1-3. Since there are 2^{mn} possible combinations of the Y's, this is a minimum total of $2^{nm} \cdot k^{nm} \cdot 2^{m(n-1)} \cdot m$ or $m2^{(2mn-m)}\, k^{nm}$ calculations for one demodulator at one set of correlation coefficients and signal-to-noise ratio. For $n = 2$ and $m = 6$, this is $1,572,864\, k^{nm}$ calculations. It is easy to see that numerical integration is practical only in trivial, non-interesting cases.

Kahn [K1] points out that Monte Carlo simulation offers a way to considerably reduce the number of calculations to evaluate expressions such as equation 2-3. In the Monte Carlo simulation, a number, N, of sets of samples of n_{ir} and Y_i, $i = 1, \ldots, n$, $r = 1, \ldots, m$, are picked according to the probability density function f_n and probabilities f_Y. Using this set of samples, the absolute error, $|e_j|$, is calculated by use of the equations developed in Chapter 1. Denoting by $|e_j|_k$ the absolute error corresponding to the kth set of samples, the MAE can be estimated by

$$\hat{MAE}_1 = \frac{1}{N} \sum_{k=1}^{N} |e_j|_k \tag{2-4}$$

Since the expected value of each $|e_j|_k$ is the MAE, this estimate is unbiased. A measure of the accuracy of the estimate is given by the variance, V, which is given by

$$\begin{aligned}
V_{\hat{MAE}_1} &= E\{[\hat{MAE}_1 - E(\hat{MAE}_1)]^2\} \\
&= E\{(\hat{MAE}_1 - MAE)^2\} \\
&= E\{(\hat{MAE}_1)^2 - 2\hat{MAE}_1\, MAE + MAE^2\} \\
&= E\{(\hat{MAE}_1)^2\} - MAE^2 \\
&= E\left\{\frac{1}{N^2} \sum_{k=1}^{N} \sum_{\ell=1}^{N} |e_j|_k |e_j|_\ell \right\} - MAE^2 \\
&= E\left\{\frac{1}{N^2}\left(\sum_{k=1}^{N} |e_j|_k^2 + \sum_{\substack{k=1 \\ k \neq \ell}}^{N} \sum_{\ell=1}^{N} |e_j|_k |e_j|_\ell\right)\right\} - MAE^2
\end{aligned}$$

Since by the selection method sets of samples are independent from other sets

$$V_{\hat{MAE}_1} = \frac{1}{N} E\{|e_j|^2\} + \frac{N-1}{N} MAE^2 - MAE^2$$

$$V_{\hat{MAE}_1} = \frac{1}{N} [E\{|e_j|^2\} - MAE^2] \tag{2-5}$$

The standard deviation of the estimates, $\sigma_{\hat{MAE}}$, equals the square root of the variance, so σ varies according to $1/\sqrt{N}$. Consequently, any accuracy that is desired can be obtained by taking N sufficiently large.

The number of samples that must be taken for a given accuracy can be reduced by using more efficient (from the Monte Carlo simulation viewpoint) estimators. For example, the absolute error, $|e_j|$, can be estimated by

$$|\hat{e}_j| = \sum_{\ell=0}^{2^m-1} |\ell - \hat{Y}_j| f_{Y_j|z}(\ell|z(t)) \tag{2-6}$$

Since $|\hat{e}_j|$ probably has less dispersion than $|e_j|$, it should be a more efficient estimator for the Monte Carlo simulation, with

$$\hat{MAE}_2 = \frac{1}{N} \sum_{k=1}^{N} |\hat{e}_j|_k \tag{2-7}$$

Since the expected value of $|\hat{e}_j|_k$ is equal to the MAE, this estimate is also unbiased and its variance is given by

$$V_{\hat{MAE}_2} = \frac{1}{N}[E\{(\hat{MAE}_2)^2\} - (MAE)^2] \tag{2-8}$$

If the values of $f_{Y_j|z}(\ell|z(t))$ are available in the demodulator (as they will be with the optimal demodulators) then this new estimate ($\hat{MAE}_2$) is more efficient and should be used.

Kahn [K1] also points out that the required number of samples can be further reduced by "importance sampling". Smith used this technique in his work to reduce his overall computer time to a reasonable amount [S2]. In importance sampling, instead of choosing samples according to the probability density function f_n and probability f_Y, samples are selected according to a modified probability density function f^+_{Yn} $(Y_1, \ldots, Y_n, n_1, \ldots, n_n)$. Equation 2-3 becomes

$$\mathrm{MAE} = \sum_{Y_1 \epsilon U} \cdots \sum_{Y_n \epsilon U} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{|e_j| f_n(n_1, \ldots, n_n) f_Y(Y_1, \ldots, Y_n)}{f^+_{Yn}(Y_1, \ldots, Y_n, n_1, \ldots, n_n)}$$

$$\cdot f^+_{Yn}(Y_1, \ldots, Y_n, n_1, \ldots, n_n)\, dn_1, \ldots, dn_n \qquad (2\text{-}9)$$

Denoting by $\{\ \}_k$ the value of the expression in braces corresponding to the kth set of samples, the estimate of the MAE becomes

$$\hat{\mathrm{MAE}}_3 = \frac{1}{N} \sum_{k=1}^{N} \left\{ \frac{|e_j| f_n f_Y}{f^+_{Yn}} \right\}_k \qquad (2\text{-}10)$$

Each sample of the absolute error is therefore weighted by $\dfrac{f_Y\, f_n}{f^+_{Yn}}$.

It will be noted that this estimate is still unbiased, and the variance is given by

$$V_{\hat{\mathrm{MAE}}_3} = \frac{1}{N}\left[E\left\{ \left(\frac{|e_j|\, f_Y\, f_n}{f^+_{Yn}} \right)^2 \right\} - (\mathrm{MAE})^2 \right] \qquad (2\text{-}11)$$

Kahn points out that if

$$f^+_{Yn} = \frac{|e_j|\, f_Y\, f_n}{\mathrm{MAE}}$$

then the variance is zero and only one sample is needed to accurately determine the MAE. However, finding this optimal f^+_{Yn} requires knowing in advance the quantity, MAE, which we want to determine, but this shows that theoretically it is possible to select f^+_{Yn} such that the variance of the estimate is reduced considerably.

If importance sampling can be used without incurring large increases in program complexity and computer time to compute each sample value of the absolute error, then it can be a valuable aid in the efficient estimation of the MAE by Monte Carlo methods. It should be noted that $|\hat{e}_j|$ can be used in place of $|e_j|$ in equations 2-10 and 2-11 when it is available.

Calculation of the variance requires knowledge of the quantity we are trying to estimate. However, the variance can be estimated by

$$\hat{V}_{\hat{MAE}_1} = \frac{1}{N}\left[\frac{1}{N}\sum_{k=1}^{N}|e_j|_k^2 - (\hat{MAE}_1)^2\right] \qquad (2\text{-}12)$$

where $|\hat{e}_j|_k^2$, $\left(\frac{|e_j|\, f_Y f_n}{f_{Yn}^+}\right)_k^2$, or $\left(\frac{|\hat{e}_j|\, f_Y f_n}{f_{Yn}^+}\right)_k^2$ is used in place of $|e_j|_k^2$ depending on which estimate we need the variance of.

These equations can be summarized as

$$AE_1 = |e_j| = |Y_j - \hat{Y}_j| \qquad (2\text{-}13a)$$

$$AE_2 = \sum_{\ell=0}^{2^m-1} |\ell - \hat{Y}_j| f_{Y_j|z}(\ell|z(t)) \qquad (2\text{-}13b)$$

$$AE_3 = AE_1 \frac{f_Y f_n}{f_{Yn}^+} \qquad (2\text{-}13c)$$

$$AE_4 = AE_2 \frac{f_Y f_n}{f_{Yn}^+} \qquad (2\text{-}13d)$$

$$\hat{MAE}_i = \frac{1}{N}\sum_{k=1}^{N}(AE_i)_k \qquad (2\text{-}14)$$

$$\hat{V}_{\hat{MAE}_i} = \frac{1}{N}\left[\frac{1}{N}\sum_{k=1}^{N}(AE_i)_k^2 - (\hat{MAE}_i)^2\right] \qquad (2\text{-}15)$$

and the parallel MSE equations by

$$SE_1 = (Y_j - \hat{Y}_j)^2 \qquad (2\text{-}16a)$$

$$SE_2 = \sum_{\ell=0}^{2^m-1} (\ell - \hat{Y}_j)^2 f_{Y_j|z}(\ell|z(t)) \qquad (2\text{-}16b)$$

$$SE_3 = SE_1 \frac{f_Y f_n}{f^+_{Yn}} \tag{2-16c}$$

$$SE_4 = SE_2 \frac{f_Y f_n}{f^+_{Yn}} \tag{2-16d}$$

$$\hat{MSE}_i = \frac{1}{N} \sum_{k=1}^{N} (SE_i)_k \tag{2-17}$$

$$\hat{V}_{\hat{MSE}_i} = \frac{1}{N}\left[\frac{1}{N} \sum_{k=1}^{N} (SE_i)_k^2 - (\hat{MSE}_i)^2 \right] \tag{2-18}$$

Due to difficulties in evaluating the MAE and MSE of the demodulators investigated by analytical or numerical methods, a Monte Carlo simulation was used to estimate these quantities. The specific way of doing this for the optimal demodulators is discussed in the next chapter. Monte Carlo methods were also used for two of the suboptimal demodulators considered and are discussed in Chapter 5.

CHAPTER 3

COMPUTER SIMULATION OF THE OPTIMAL DEMODULATORS

Equations 1-1, 1-2, 1-3 and 1-13 involve operations ideally suited to digital computers. As pointed out in Chapter 2, a Monte Carlo simulation appears to offer the best approach to the evaluation of the MAE and MSE for the three optimal demodulators that we are considering.

To accomplish this simulation, the data samples, $x_1^*, \ldots, x_n^*$, are picked from the proper distribution, quantized into the digital signals, $Y_1, \ldots, Y_n$, and the Y_i are then coded into binary PCM signals. Noise samples are selected from the proper noise distribution and added to the PCM signals to form the received signal. The 2^m values of $f_{Y_j|z}$ are calculated by use of equation 1-13, and these are used in equations 1-1, 1-2, and 1-3 to get $\hat{Y}_j$ for each demodulator. Equations 2-13 and 2-16 are used to calculate the estimators for the absolute error and the square error. This procedure is repeated N times and equations 2-14 and 2-17 are used to calculate the $\hat{\text{MAE}}$ and $\hat{\text{MSE}}$. The accuracy of these estimates can then be estimated from equations 2-15 and 2-18.

As was pointed out in the introduction, the data samples are assumed to be normally distributed, with a specified correlation between samples. The following procedure was used to generate the data samples on the computer. For n = 2, i.e., two words considered at a time by the demodulator, the demodulator estimates Y_2 using z_1 and z_2. x_1^* is picked from a normal distribution[1] with mean m_Y and standard

[1] Most large computer installations have random number generators in their libraries, both for uniform and normal distributions. Such was the case at the University of Michigan Computer Center where the simulation of the optimal and suboptimal demodulators was done.

deviation σ_Y. Then x_2^* is picked from the conditional probability distribution of x_2^* given x_1^*. For joint normal data samples with correlation coefficient ρ, the conditional probability density function is given by

$$f_{x_2|x_1}(x_2^*|x_1^*) = \frac{f_X(x_1^*, x_2^*)}{f_X(x_1^*)}$$

$$= \frac{\dfrac{1}{2\pi\sigma_Y^2\sqrt{1-\rho^2}} \exp\left\{-\dfrac{(x_1^* - m_Y)^2 - 2\rho(x_1^* - m_Y)(x_2^* - m_Y) + (x_2^* - m_Y)^2}{2\sigma_Y^2(1-\rho^2)}\right\}}{\dfrac{1}{\sqrt{2\pi}\,\sigma_Y} \exp\left\{-\dfrac{(x_1^* - m_Y)^2}{2\sigma_Y^2}\right\}}$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma_Y\sqrt{1-\rho^2}} \exp\left\{-\frac{[x_2^* - (m_Y - \rho m_Y + \rho x_1^*)]^2}{2\sigma_Y^2(1-\rho^2)}\right\} \qquad (3\text{-}1)$$

Therefore, x_2^* is selected from a normal distribution having mean $= m_Y - \rho m_Y + \rho x_1^*$ and standard deviation $= \sigma_Y\sqrt{1-\rho^2}$.

For n = 3, the demodulator estimates Y_2 given z_1, z_2, and z_3.[1] x_2^* is selected from a normal (m_Y, σ_Y) distribution, and x_1^* and x_3^* are then selected from the normal distribution having a mean of $m_Y - \rho m_Y + \rho x_2$ and a standard deviation of $\sigma_Y\sqrt{1-\rho^2}$. The correlation coefficient between x_1^* and x_2^* and between x_2^* and x_3^* is ρ. The selection method fixes the correlation coefficient between x_1^* and x_3^*, ρ_{13}, and it is determined as follows.

$$\rho_{13} = \frac{E[x_1^* x_3^*] - m_Y^2}{\sigma_Y^2}$$

Now

$$E[x_1^* x_3^*] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1^* x_3^* f_X(x_1^*, x_3^*)\, dx_1^*\, dx_3^*$$

[1] In the majority of the applications of PCM demodulation there will be no constraint to operate without a time delay.

Since

$$f_X(x_1^*, x_3^*) = \int_{-\infty}^{\infty} f_X(x_1^*, x_2^*, x_3^*)\, dx_2^*$$

and

$$f_X(x_1^*, x_2^*, x_3^*) = f_{X|YZ}(x_1^*|x_2^*, x_3^*)\, f_{X|Y}(x_3^*|x_2^*)\, f_X(x_2^*)$$

$$= f_{X|Y}(x_1^*|x_2^*)\, f_{X|Y}(x_3^*|x_2^*)\, f_X(x_2^*)$$

then

$$E[x_1^*\ x_3^*] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1^* f_{X|Y}(x_1^*|x_2^*)\, dx_1^* \int_{-\infty}^{\infty} x_3^* f_{X|Y}(x_3^*|x_2^*)\, dx_3^*\, f_X(x_2^*)\, dx_2^*$$

$$= \int_{-\infty}^{\infty} E[x_1^*|x_2^*]\, E[x_3^*|x_2^*]\, f_X(x_2^*)\, dx_2^*$$

$$= \int_{-\infty}^{\infty} (m_Y(1 - \rho) + \rho x_2^*)^2 f_X(x_2^*)\, dx_2^*$$

$$= m_Y^2(1 - 2\rho + \rho^2) + 2m_Y(1 - \rho)\rho \int_{-\infty}^{\infty} x_2^* f(x_2^*)\, dx_2^*$$

$$+ \rho^2 \int_{-\infty}^{\infty} x_2^{*2} f(x_2^*)\, dx_2^*$$

$$= m_Y^2\ (1 - 2\rho + \rho^2) + 2m_Y^2(\rho - \rho^2) + \rho^2(\sigma_Y^2 + m_Y^2)$$

$$= m_Y^2 + \rho^2\sigma_Y^2$$

Consequently

$$\rho_{13} = \frac{\rho^2\sigma_Y^2}{\sigma_Y^2} = \rho^2$$

The extension to higher values of n is obvious, but is not considered here because of the excessive computer time required to simulate the demodulation for $n > 3$, as will be obvious in Chapter 4.

Two different size PCM words are considered, three-bit words ($m = 3$) and six-bit words ($m = 6$). For six-bit words, the data samples are quantized according to figure 3-1a. Data samples outside the range $m_Y \pm 2.6\,\sigma_Y$ are assigned values of 0 or 63 and the interior range is divided into equal parts corresponding to the signals 1 through 62. For six-bit words then,

$$
\begin{aligned}
m_Y &= \frac{2^m - 1}{2} = 31.5 \\
\sigma_Y &= \frac{m_Y}{2.6} = \frac{31.5}{2.6} = 12.115
\end{aligned}
\tag{3-2}
$$

For three-bit words, the first three bits of the six-bit words are used. Figure 3-1b results, with

$$
\begin{aligned}
m_Y &= 3.5 \\
\sigma_Y &= \frac{3}{1.9803} = 1.515
\end{aligned}
\tag{3-3}
$$

The signal power per bit time, S^2, is given by

$$
S^2 = \frac{1}{T_B} \int_{T_B} g_1^2(t)\,dt = \frac{1}{T_B} \int_{T_B} g_2^2(t)\,dt \tag{3-4}
$$

A time average correlation, λ, between $g_1(t)$ and $g_2(t)$ is defined by

$$
\lambda = \frac{1}{T_B S^2} \int_{T_B} g_1(t)\, g_2(t)\,dt \tag{3-5}
$$

The root-mean-square (RMS) signal-to-noise ratio, S/N, is defined as the square root of the ratio of the signal power to the noise power in a bandwidth equal to the bit rate, i.e.,

$$
S/N = \sqrt{\frac{S^2}{\sigma_n^2 / T_B}} = \frac{\sqrt{T_B}\, S}{\sigma_n} \tag{3-6}
$$

Consider the exponential expression in equation 1-13,

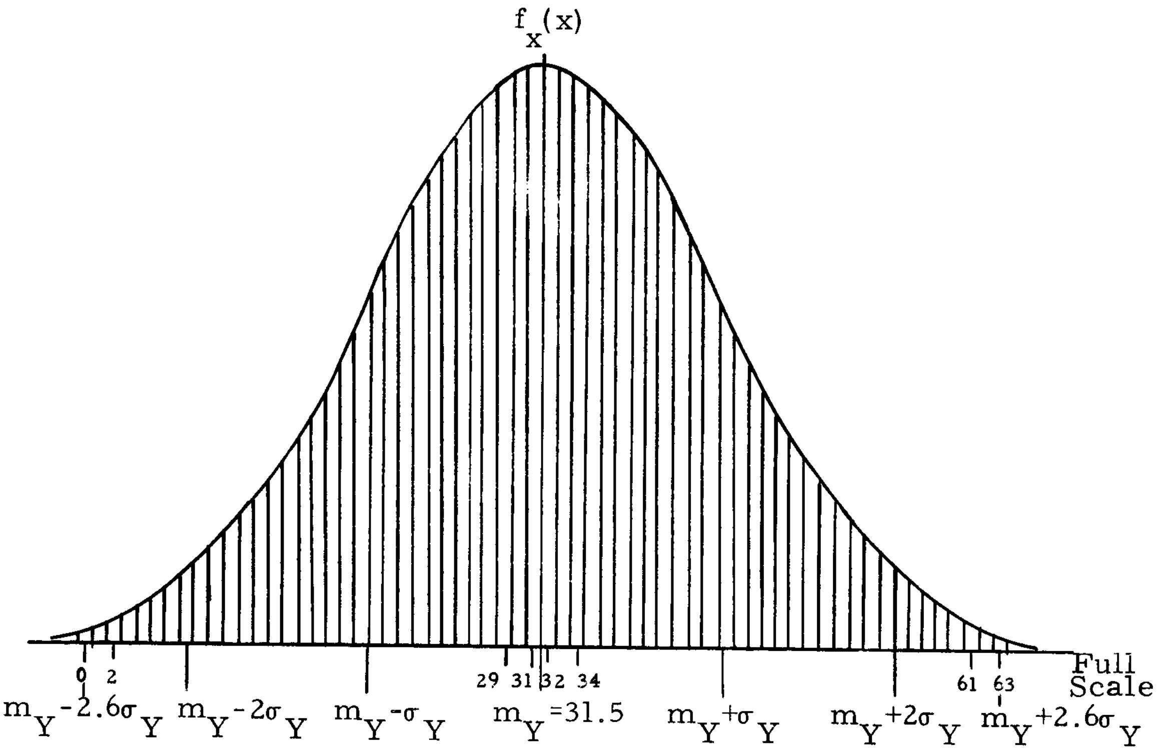

Figure 3-1a. Quantization Intervals, 6-bit Words

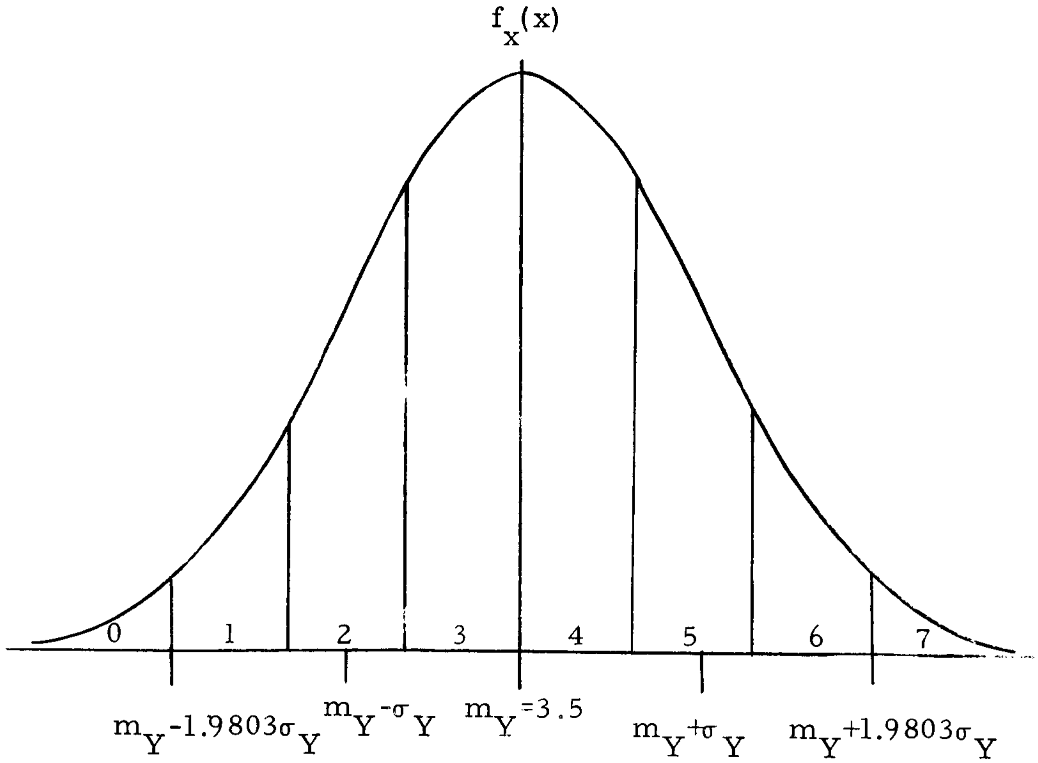

Figure 3-1b. Quantization Intervals, 3-bit Words

$$\exp\left\{\frac{1}{\sigma_n^2}\sum_{r=1}^{m}\int_{T_B} z_{ir}(t)y_{i(\ell)r}(t)\,dt\right\} \tag{3-7}$$

This can be expanded to

$$\exp\left\{\frac{1}{\sigma_n^2}\sum_{r=1}^{m}\int_{T_B} (y_{ir}(t)+n_{ir}(t))\,y_{i(\ell)r}(t)\,dt\right\}$$

where $n_{ir}(t)$ is the noise waveform during the bit time, T_B.

$$= \exp\left\{\sum_{r=1}^{m}\left[\frac{1}{\sigma_n^2}\int_{T_B} y_{ir}(t)y_{i(\ell)r}(t)\,dt + \frac{1}{\sigma_n^2}\int_{T_B} n_{ir}(t)y_{i(\ell)r}(t)\,dt\right]\right\} \tag{3-8}$$

Consider the first part of the expression in the brackets in the above equation.

$$\frac{1}{\sigma_n^2}\int_{T_B} y_{ir}(t)y_{i(\ell)r}(t)\,dt = \begin{cases} \dfrac{T_B S^2}{\sigma_n^2}\,, & \text{for matched bits} \\[2ex] \lambda\,\dfrac{T_B S^2}{\sigma_n^2}\,, & \text{for unmatched bits} \end{cases}$$

The second part of the expression in brackets in equation 3-8 is a random variable since it is the integral of a random process, $n_{ir}(t)$. This random variable has mean 0 and variance:

$$V = \frac{1}{\sigma_n^4}\int_{T_B}\int_{T_B} E\{n_{ir}(t)y_{i(\ell)r}(t)n_{ir}(\tau)y_{i(\ell)r}(\tau)\}\,dt\,d\tau$$

$$= \frac{1}{\sigma_n^4}\int_{T_B}\int_{T_B} y_{i(\ell)r}(t)y_{i(\ell)r}(\tau)\;E\{n_{ir}(t)n_{ir}(\tau)\}\,dt\,d\tau$$

$$= \frac{1}{\sigma_n^4}\int_{T_B}\int_{T_B} y_{i(\ell)r}(t)y_{i(\ell)r}(\tau)\;\sigma_n^2\,\delta(t-\tau)\,dt\,d\tau$$

$$= \frac{T_B S^2}{\sigma_n^2}$$

Therefore, the expression in brackets in equation 3-8 reduces to

$$\frac{1}{\sigma_n^2} \int_{T_B} y_{i(\ell)r}(t)\, z_{ir}(t)\, dt = \begin{cases} (S/N)^2 + J, & \text{matched bits} \\ \lambda (S/N)^2 + J, & \text{unmatched bits} \end{cases}$$

where J is a random variable with mean 0 and variance $(S/N)^2$. Smith [S2] points out that the computer simulation can be accomplished most simply by using $g_1(t) = +S/N$ and $g_2(t) = -S/N$ and selecting noise samples, denoted by ν_{ir}, from a normal (0, 1) distribution. For these PCM waveforms $\lambda = -1$. The generalization to arbitrary waveforms and arbitrary λ is given in Appendix II. Equation 3-7 becomes

$$\exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{ir} y_{i(\ell)r} dt\right\} = \exp\left\{\sum_{r=1}^{m} (y_{ir} + \nu_{ir}) y_{i(\ell)r}\right\} \qquad (3\text{-}9)$$

where $y_{\cdot(\cdot)\cdot}$ equals either a plus or minus S/N and ν_{ir} is normal (0,1).

Therefore, after the quantization of x_i^* into Y_i, Y_i is coded into a series of ± 1's. These are multiplied by S/N to form the y_{ir}'s, and to each of these products is added a sample from a Gaussian distribution with mean 0 and standard deviation 1. This forms the quantity in parenthesis in the right side of equation 3-9, and the computer has all the information that is needed to start the demodulation.

For the demodulation, $f_Y(\ell_1, \ldots, \ell_{j-1}, k, \ell_{j+1}, \ldots, \ell_n)$ is required. For samples from a Gaussian distribution that we are considering

$$f_Y(\ell_1, \ldots, \ell_{j-1}, k, \ell_{j+1}, \ldots, \ell_n)$$
$$= \int_{R_{\ell_1}} \cdots \int_{R_{\ell_{j-1}}} \int_{R_k} \int_{R_{\ell_{j+1}}} \cdots \int_{R_{\ell_n}} f_X(x_1, \ldots, x_n)\, dx_1 \cdots dx_n$$

where f_x is the multivariate Gaussian density function and R_{ℓ_i} is the region where x_i quantizes into ℓ_i. For the computer simulation, sufficient accuracy is obtained by taking the value of f_x at the midpoint of these regions, i.e.,

$$f_Y = f_x(\ell_1, \ldots, \ell_{j-1}, k, \ell_{j+1}, \ldots, \ell_n)$$

For n = 2,

$$f_Y(\ell_1, k) = K_4 \exp\left\{- \frac{(\ell_1 - m_Y)^2 - 2\rho(\ell_1 - m_Y)(k - m_Y) + (k - m_Y)^2}{2\sigma_Y^2(1 - \rho^2)}\right\} \tag{3-10}$$

For n = 3, x_1^* and x_3^* were chosen to have a correlation coefficient ρ with x_2^*. As shown before the correlation coefficient between x_1^* and x_3^* is ρ^2. For this case

$$\begin{aligned} f_x(x_1, x_2, x_3) &= f_{x_1|x_2,x_3}(x_1|x_2, x_3)\, f_x(x_2, x_3) \\ &= f_{x_1|x_2}(x_1|x_2)\, f_x(x_2, x_3) \\ &= \frac{f_x(x_1, x_2)\, f_x(x_2, x_3)}{f_x(x_2)} \end{aligned}$$

Therefore

$$f_Y(\ell_1, k, \ell_3) = \frac{f_Y(\ell_1, k)\, f_Y(k, \ell_3)}{f_Y(k)} \tag{3-11}$$

where $f_Y(\ell_1, k)$ and $f_Y(k, \ell_3)$ are given by equation 3-10 and

$$f_Y(k) = K_5 \exp\left\{- \frac{(k - m_Y)^2}{2\sigma_Y^2}\right\} \tag{3-12}$$

It should be noted here that samples from distributions other than a Gaussian distribution could be used in the computer simulation. As mentioned in the introduction, a way of generating correlated samples from the new distribution must be available for the computer. Most large computers have available subroutines for generating pseudorandom numbers from a uniform or a Gaussian distribution. Kahn [K2] discusses ways of using uniformly distributed random numbers to generate samples from other distributions. In addition to generating samples from the new distribution, a method must be available for calculating $f_Y(\ell_1, \ldots, \ell_{j-1}, k, \ell_{j+1}, \ldots, \ell_n)$ for use in the demodulator.

For small signal-to-noise ratios, the Monte Carlo simulation gave reasonably accurate estimates of the MAE and MSE for the optimal demodulators with a reasonable number of samples (and hence reasonable lengths of computer time). However, for S/N larger than 1.414, the number of samples (and computer time) required for reasonable accuracy became excessive. Importance sampling was used to improve the convergence of the estimation and therefore, reduce the computation time to a tractable amount.

As pointed out in Chapter 2, the modified sampling distribution, f^+_{Yn}, should be taken proportional to $|e_j| f_Y f_n$ or $(e_j)^2 f_Y f_n$. These are very complicated functions due to the complicated nature of the optimal demodulators, but some indication of the approach to take to determine a good but simple f^+_{Yn} can be gained by considering a simple example of one-bit words taken two at a time ($m = 1, n = 2$). For this case Y_1 and Y_2 can take on the values of 0 or 1. Letting ν_1 and ν_2 take on various values, $SE_2 f_n$ was plotted for the different combinations of Y_1 and Y_2. Figure 3-2a shows the results for $Y_1 = 0$ and $Y_2 = 0$ and figure 3-2b shows the case where $Y_1 = 1$, $Y_2 = 0$. For both Y_1 and Y_2 equal to 1 the plot is the same as figure 3-2a except reversed so the peak occurs at approximately $\nu_1 = -S/N$, $\nu_2 = -S/N$. For $Y_1 = 0$, $Y_2 = 1$ the reverse of figure 3-2b occurs.

From these plots, it was decided to pick the noise samples from a normal distribution with mean B and standard deviation A. This noise sample ν_{ir} was then given a sign opposite to that of the PCM word bit to which it was added. This simple scheme approximates to

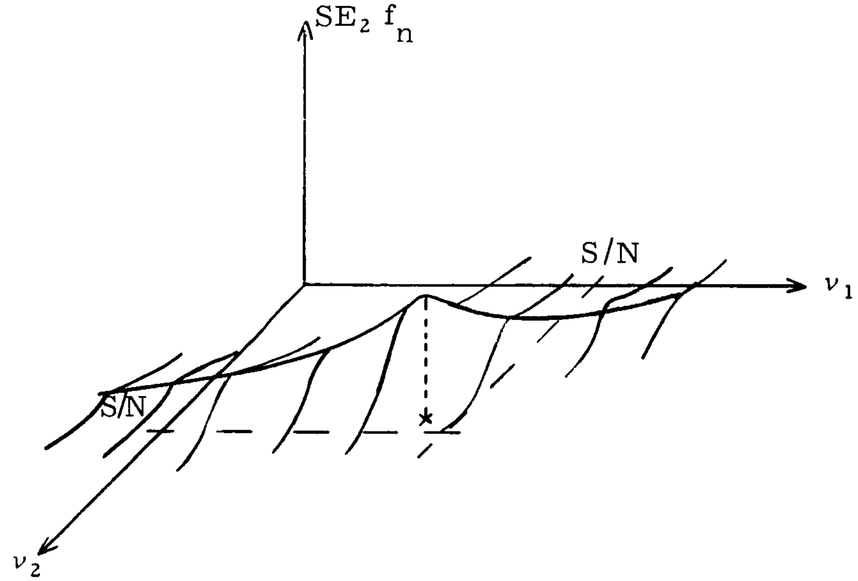

Figure 3-2a. $SE_2 f_n$ for m = 1, n = 2, $Y_1 = Y_2 = 0$

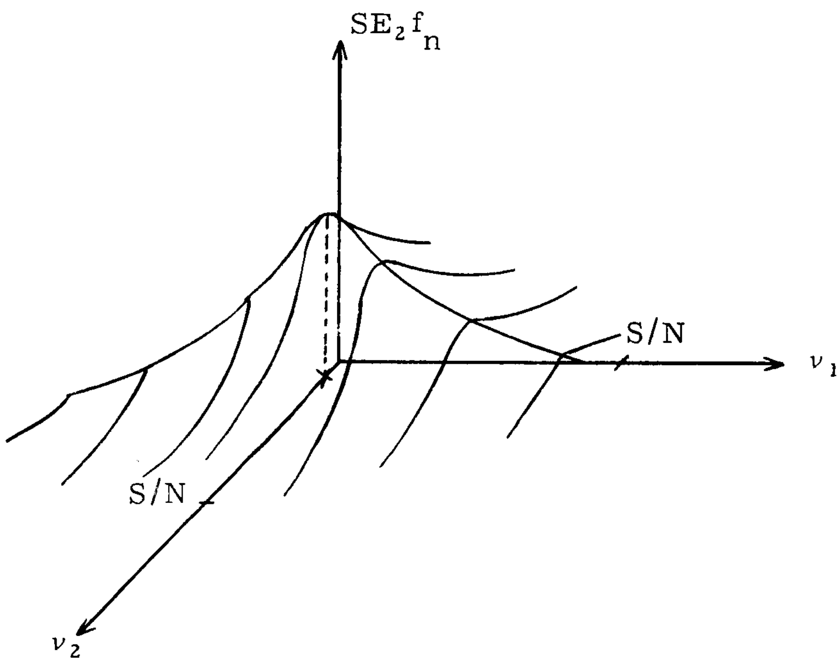

Figure 3-2b. $SE_2 f_n$ for m = 1, n = 2, $Y_1 = 1$, $Y_2 = 0$

some degree the plots in figure 3-2, yet can be accomplished in the computer simulation with a minimum of program complexity and extra running time.

Denoting the weighting of each sample by G, we have

$$G = \frac{f_n f_Y}{f^+_{Yn}} = \frac{f_n f_Y}{f^+_{n|Y} f^+_Y} = \frac{f_n}{f^+_{n|Y}}$$

since $f^+_Y = f_Y$. The conditional distribution of n given Y is a normal distribution centered at plus or minus B, depending on whether y_{ir} is positive or negative. Taking ν_{ir} as the value of the noise sample before the sign is assigned, then

$$f^+_{n|Y} = \prod_{i=1}^{n} \prod_{r=1}^{m} \frac{1}{\sqrt{2\pi}\, A} \exp\left\{-\frac{(\nu_{ir} - B)^2}{2A^2}\right\}$$

and

$$G = \prod_{i=1}^{n} \prod_{r=1}^{m} \frac{\frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(\nu_{ir})^2}{2}\right\}}{\frac{1}{\sqrt{2\pi}\, A} \exp\left\{-\frac{(\nu_{ir} - B)^2}{2A^2}\right\}}$$

$$= A^{nm} \exp\left\{\frac{mnB^2}{2A^2}\right\} \exp\left\{\frac{(1 - A^2) \sum_{r=1}^{m} \sum_{i=1}^{n} \nu_{ir}^2 - 2B \sum_{r=1}^{m} \sum_{i=1}^{n} \nu_{ir}}{2A^2}\right\} \tag{3-13}$$

The best value of A and B were determined for each ρ and S/N by trial simulations on the computer. The use of this scheme of importance sampling gave a reduction of the number of sample sets required for a specified accuracy by a factor of up to 4.

The computer program for the optimal demodulator simulation is discussed in Table A-3-2 of Appendix III. The results of the simulation are discussed in the next chapter.

CHAPTER 4

OPTIMAL DEMODULATOR PERFORMANCE

The quantization error for 3 and 6-bit data is developed in Appendix I. Correlation coefficients for Butterworth data were also determined in this Appendix for 1 % (6-bit) data and 10% (3-bit) data (see Table A-1-1). Based on these correlation coefficients, for n = 2 (two words) it was decided to run the Monte Carlo simulation of the optimal demodulators on the digital computer for correlation coefficients of .95, .98, and .995 for 6-bit words and ρ's of .9, .95, and .98 for 3-bit words. S/N's starting at .707 and going up to those that gave MAE's or RMSE's near the quantization error were used, since the performance is limited by the quantization error. This meant going up to a S/N of 2.83 for 6-bit words and a S/N of 2.0 for 3-bit words.

Some data was also run off for the three word case (n = 3), however due to the very long running time on the computer for this program, this was only done for 3-bit words. Two runs were made for each S/N, at correlation coefficients of .95 and .98.

For comparison, the MAE and RMSE for the optimal demodulators was computed for the case where only one word was used in the demodulation process (n = 1), for the same S/N's used in the two and three word case. These values were also used to estimate the demodulator performance for ρ = 1, as discussed in the third paragraph below.

There was a necessary compromise between the desired accuracy of the results and the available computer time. Although the simulation was done on the University of Michigan IBM 7090 Computer

System, a comparatively high speed computer, for feasible running times on the computer it was necessary to limit the accuracy to having the estimate of two standard deviations (equations 2-15 and 2-18) be less than 10% of the estimated value of the MAE and MSE. Since the $\hat{\text{MAE}}$ and $\hat{\text{MSE}}$ have an asymptotically normal distribution[1], we will then have a 95% confidence level that the true result is within 10% of the estimated value. This was slipped to around 20% for a S/N of 2.83 due to the extremely large number of iterations required. The required computer times are given in Table A-4-8 of Appendix IV.

The optimal demodulator results for the 1, 2, and 3 word cases are given in Tables A-4-2a, A-4-3a, and A-4-4a of Appendix IV. Included in these tables are the number of iterations required (N) and the values of A and B where importance sampling was used. The values of $2\hat{\sigma}$ are given for the $\hat{\text{MAE}}$ and $\hat{\text{RMSE}}$ (for the $\hat{\text{RMSE}}$, $2\hat{\sigma}_{\hat{\text{MSE}}}$ was used to compute the high and low values of the $\hat{\text{RMSE}}$, and the largest deviation from the $\hat{\text{RMSE}}$ was considered as $2\hat{\sigma}_{\hat{\text{RMSE}}}$). The results ($\hat{\text{MAE}}$, $\hat{\text{RMSE}}$, $2\hat{\sigma}$) are normalized by dividing by the peak-to-peak signal (63 for 6-bit words, 7 for 3-bit words).

For $\rho = 1$, equation 1-13 reduces to

$$f_{Y_j|Z}(k|z(t)) = K_3(z)\, f_{Y_j}(k)\, \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} \sum_{i=1}^{n} z_{ir} y_{j(k)r}\, dt\right\}$$

since $f_Y(\ell_1, \ldots, \ell_{j-1}, k, \ell_{j+1}, \ldots, \ell_n)$ equals zero unless all its arguments are identical. This is equivalent to the expression obtained for the one word case ($n = 1$) with $\sum_{i=1}^{n} z_{ir}$ substituted for z_{jr}, or an averaging of the received PCM words. For $n = 2$, there is therefore a quadrupling of the apparent signal power, while, due to the independence of the additive noise, the noise power is increased only by 2.

[1] This is the essence of the central limit theorem of probability theory [C1].

This means a power savings of 3 db when two received words are averaged. For n = 3 the power savings is 3 or 4.77 db. This allows us to plot the $\rho = 1$, $n = 2$, line 3 db to the left of the $n = 1$, optimal demodulation line on plots of the MAE or RMSE against the S/N in db. The $n = 3$, $\rho = 1$, line is, by the same reasoning, 4.77 db to the left of the $n = 1$ line on these plots.

It is also convenient for comparison purposes to plot the performance of present day PCM demodulators that use bit-by-bit correlation of the received waveforms with the known signal waveforms $g_1(t)$ and $g_2(t)$ (or $f_1(t)$ and $f_2(t)$ in the general case (see Appendix II)) to determine the estimation of the transmitted bit, $\hat{y}_{ir}$. For this type of demodulator, the probability of an error in a bit is equal to the probability that the integral of the channel noise over the bit time is greater than and of opposite sign to the integral of the signal. By the reasoning of Chapter 3, this is equivalent to the probability that a noise sample from a normal (0, 1) distribution is larger than $\pm S/N$ but with the opposite sign, values of which can be found from standard statistical tables since there is an equal probability that the transmitted bit is equal to a plus or minus S/N. The probability of a certain size error is then found by summing the product of the probability of each different combination of bit errors that will give that size error and the probability that the necessary signal was transmitted, over all possible combinations that give this size error. The MAE (or MSE) is then the sum of the probability of an error of size i times i (or i^2 for the MSE) for all i. The computer program to do this is given in Table A-3-1 of Appendix III and the results for both 3-bit and 6-bit words are given in Table A-4-1 of Appendix IV.

These results of the computer simulation for the three optimal demodulators are plotted on figures 4-1 through 4-9 vs. the S/N in db. The $\hat{\text{MAE}}$ and $\hat{\text{RMSE}}$ are normalized by the peak-to-peak (P-P) signal.

Figures 4-1 through 4-6 show the normalized $M\hat{A}E$ and $RM\hat{S}E$ for 6-bit words for each of the three optimal demodulators. The results for bit-by-bit demodulation are also shown on these plots. Figure 4.7 shows comparison of the $M\hat{A}E$ and $RM\hat{S}E$ of the three optimal demodulators for $\rho = .98$. Figure 4.8 and 4.9 show the $M\hat{A}E$ and $RM\hat{S}E$ of the minimum MAE demodulator and the minimum MSE demodulator respectively for 3-bit words. In addition to the bit-by-bit demodulation results, the results for three words ($n = 3$) are shown. The quantization error is also shown on all these graphs as a reference level.

It is evident from these graphs that the minimum P_e and minimum MAE demodulators perform similarly, while the performance of the minimum MSE demodulator has a somewhat different behavior. As pointed out before, the minimum P_e and MAE demodulators can only give discrete (i.e., a possible value of Y_2) values of $\hat{Y}_2$, while the minimum MSE demodulator gives continuous values of $\hat{Y}_2$. Also, a comparison of figures 4-3 and 4-8 and figures 4-6 and 4-9 shows that the performance of the 3-bit optimal demodulators is quite similar to the 6-bit optimal demodulators, close enough to use less expensive (in terms of computer time) 3-bit Monte Carlo simulations to predict trends in 6-bit (and longer) optimal demodulators.

It is also evident from these graphs that for high correlation coefficients, the optimal demodulators give a performance approaching the $\rho = 1$ line for the number of words considered. Improvement in performance is possible by considering more words in each demodulation, but from equation 1-13 it is obvious that an increase in n will greatly increase the demodulation time. In the simulation programs, an increase in n from 2 to 3 increased the computer running time by a factor of around four (see Table A-4-8 of Appendix IV).

Although we have been discussing PCM demodulation when the system parameters (word size, S/N, etc.) are fixed, the system

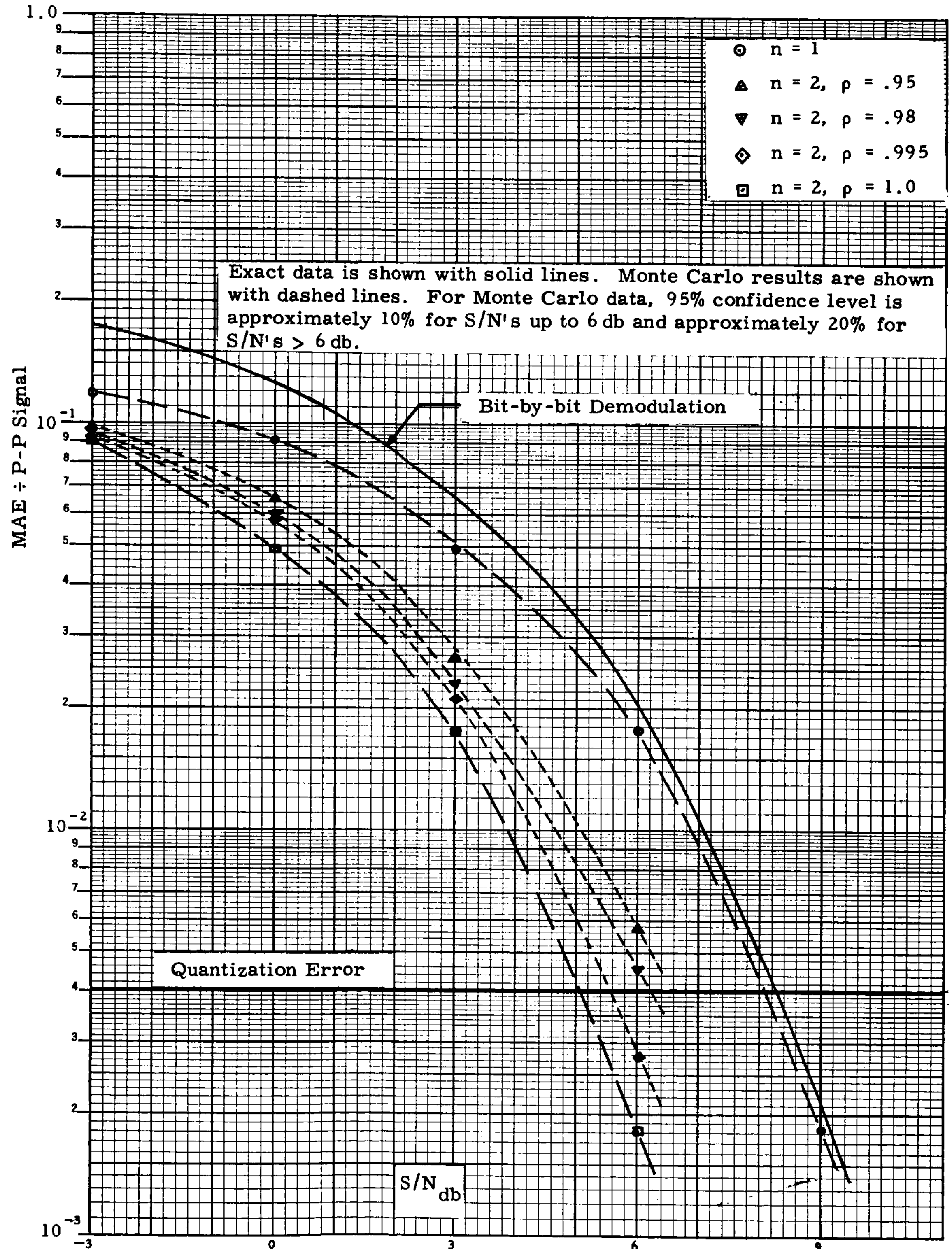

Figure 4-1. MAE for the Minimum P_e Demodulator, m = 6.

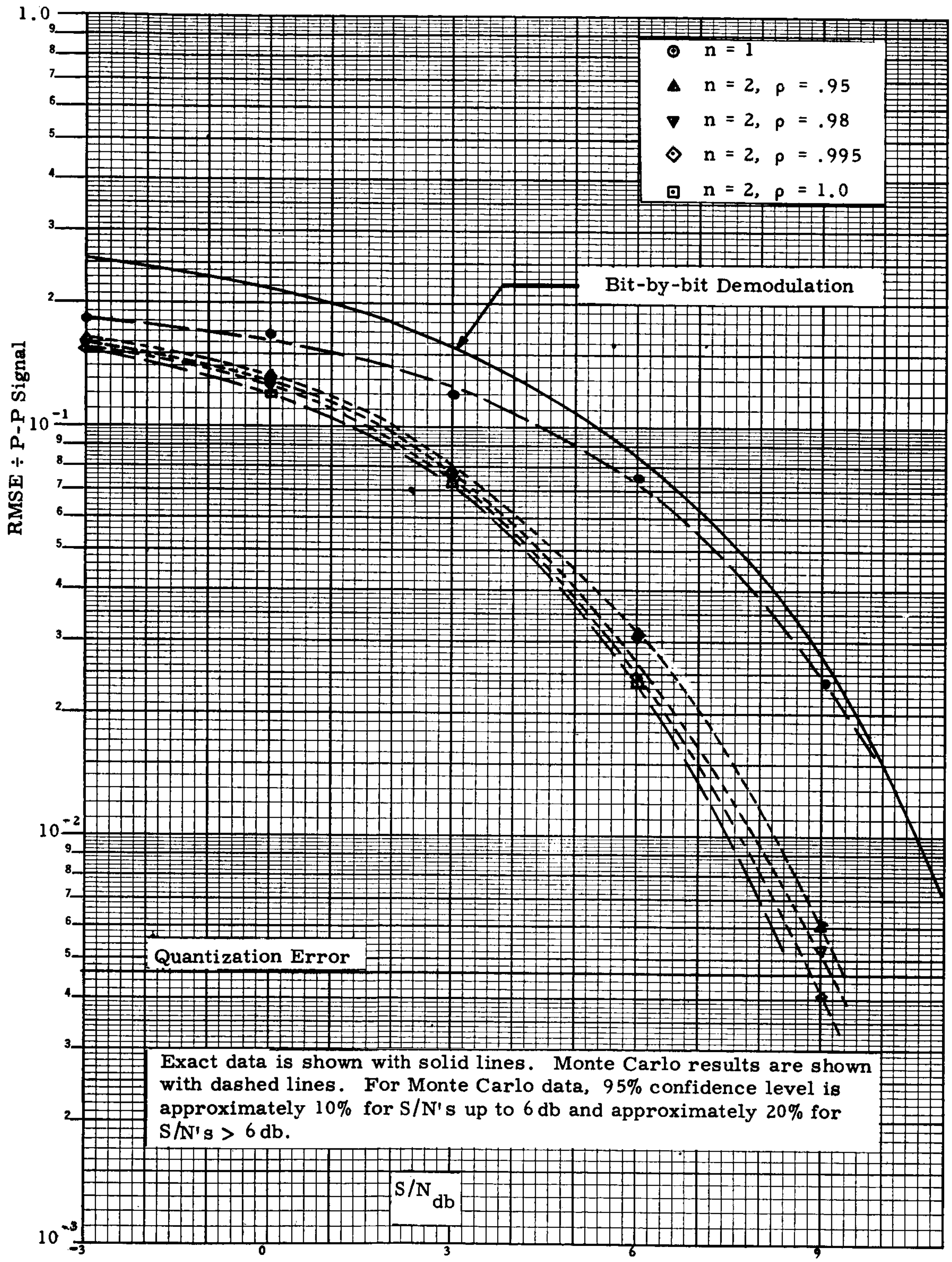

Figure 4-2. RMSE for the Minimum P_e Demodulator, m = 6.

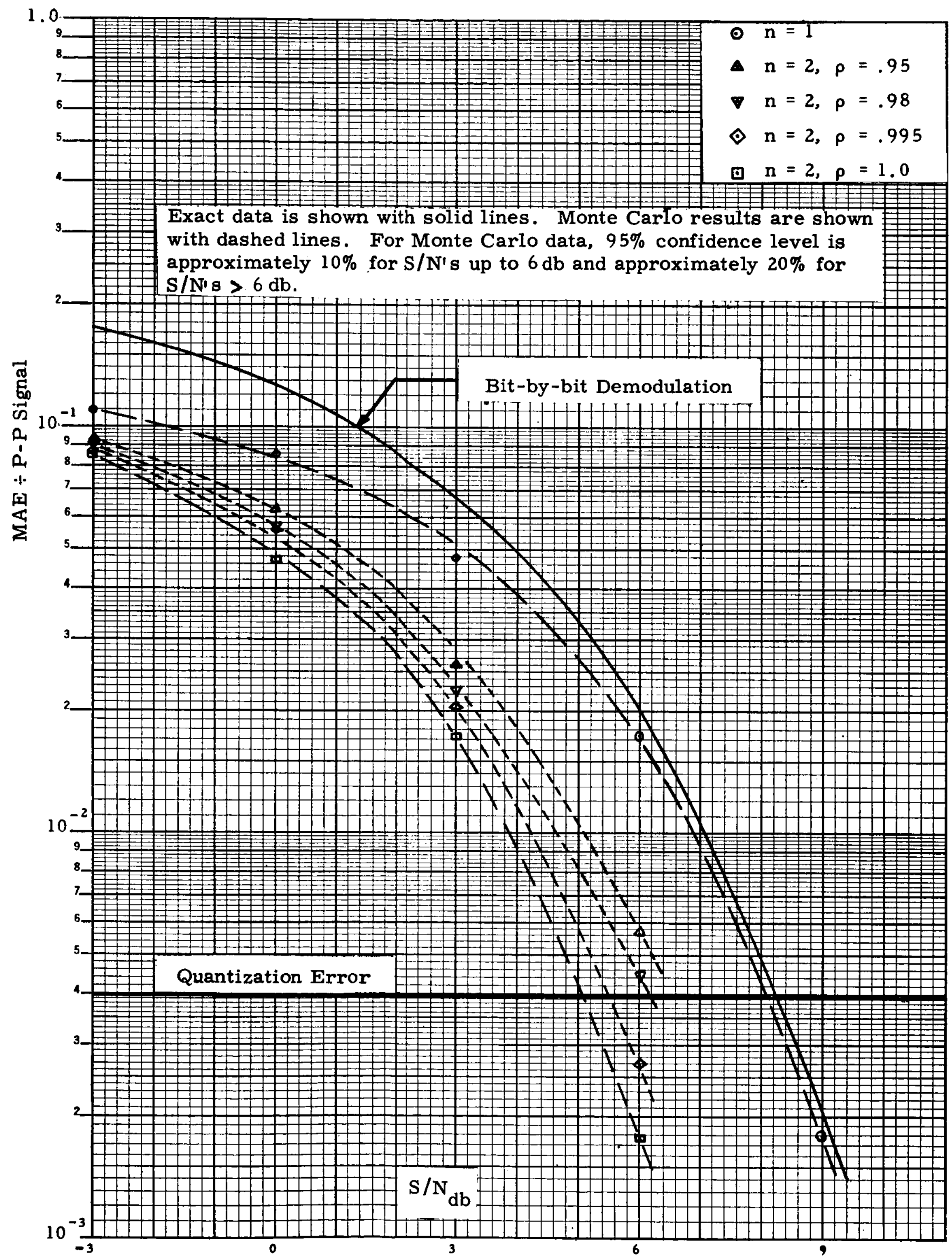

Figure 4-3. MAE for the Minimum MAE Demodulator, m = 6.

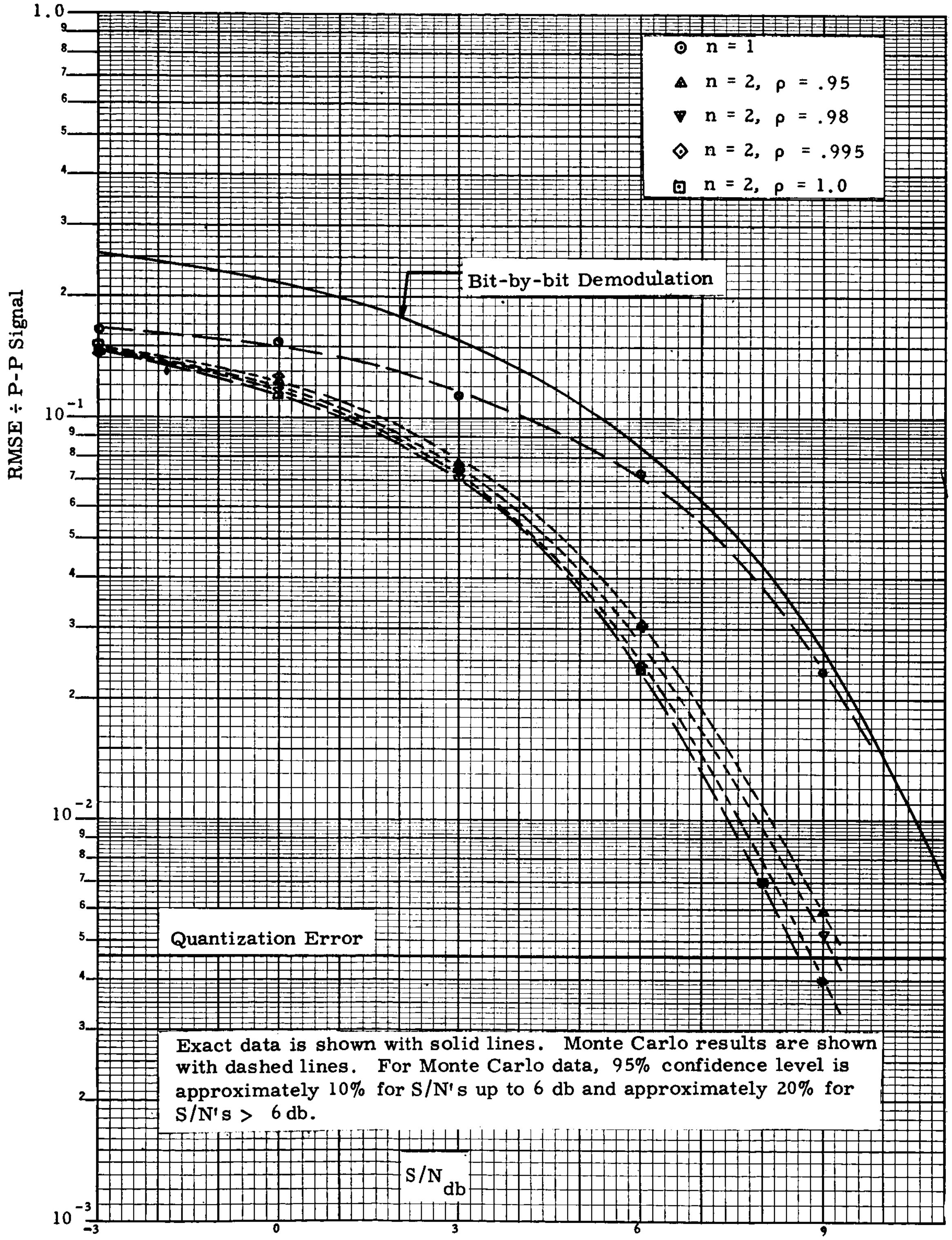

Figure 4-4. RMSE for the Minimum MAE Demodulator, m = 6.

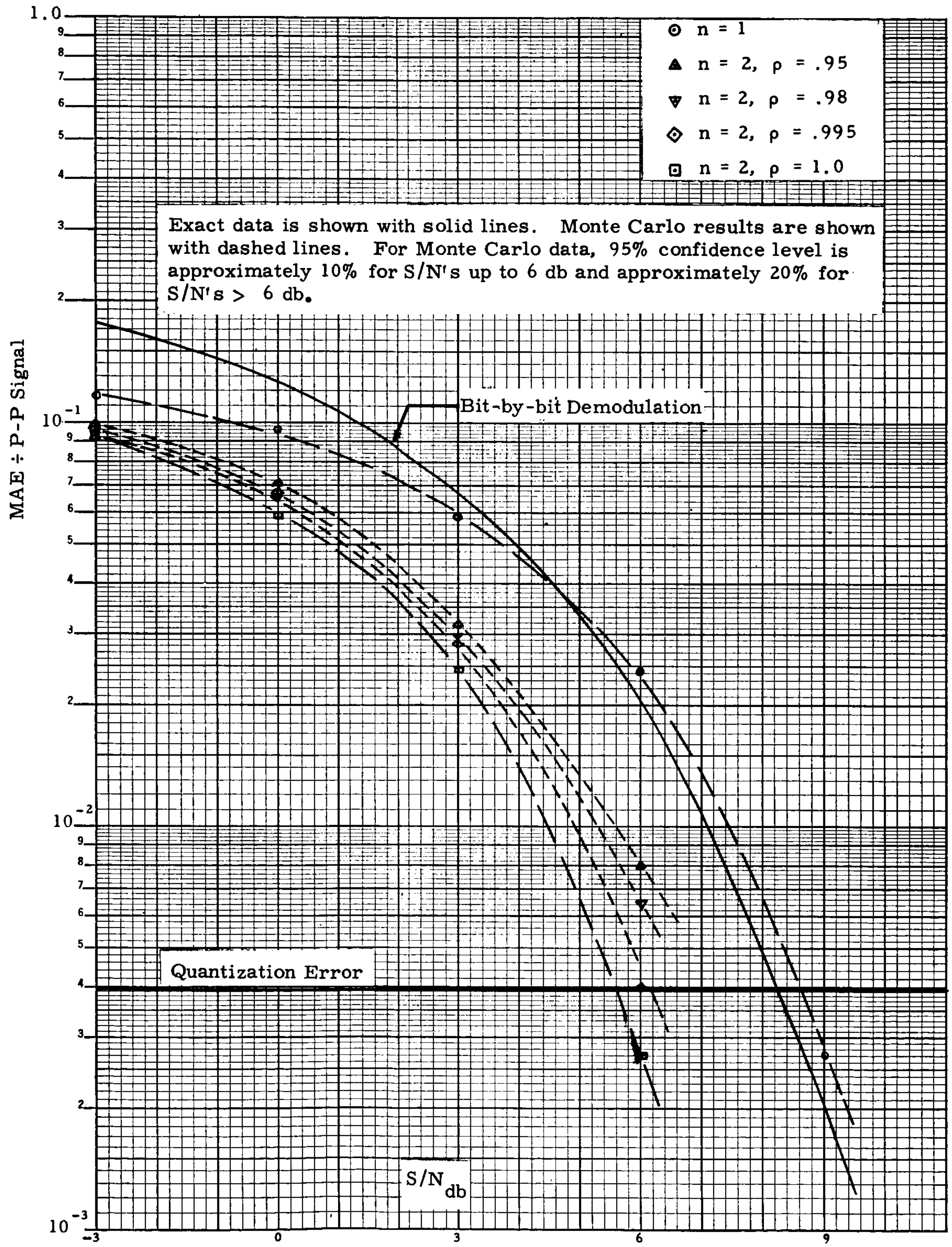

Figure 4-5. MAE for the Minimum MSE Demodulator, m = 6.

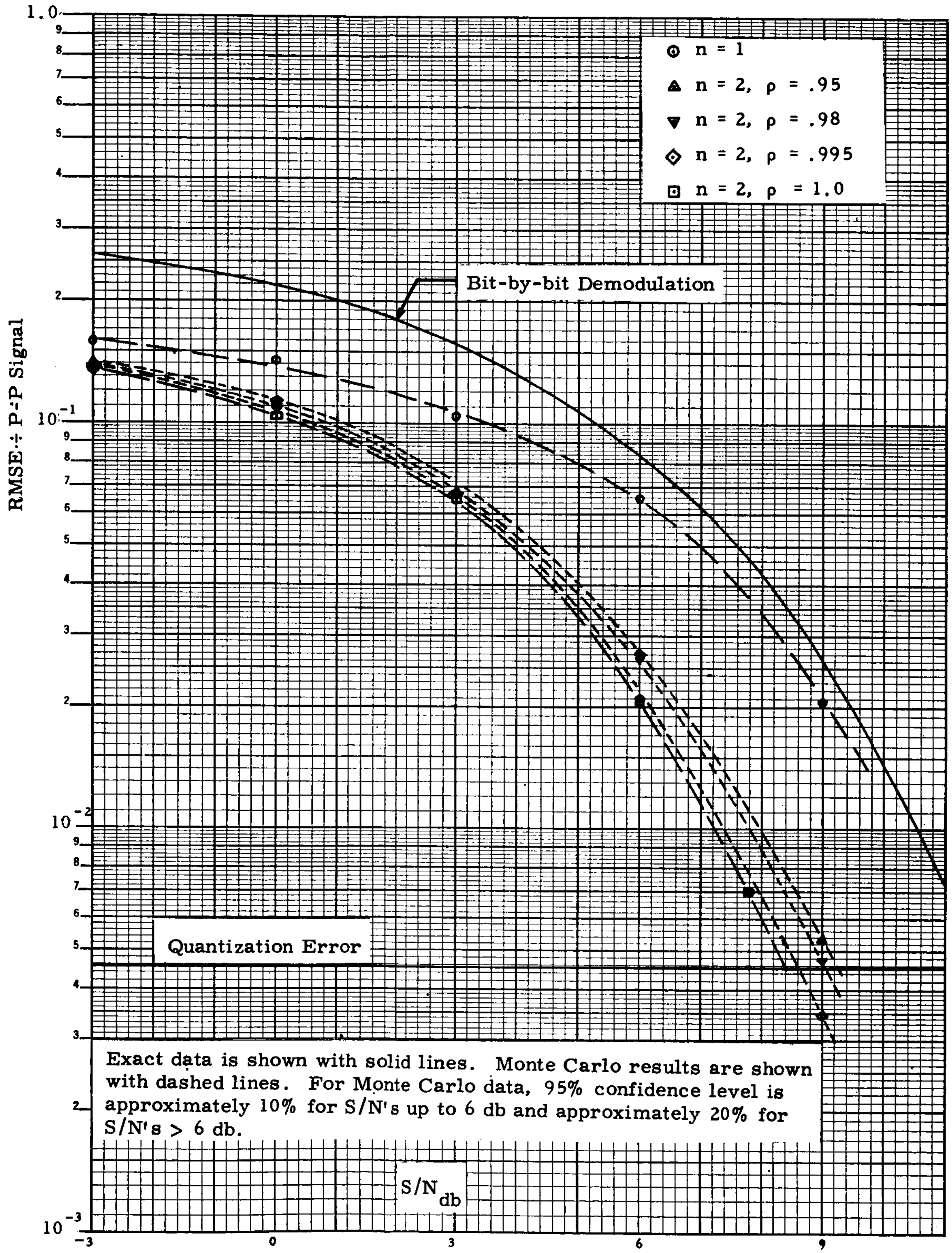

Figure 4-6. RMSE for the Minimum MSE Demodulator, m = 6.

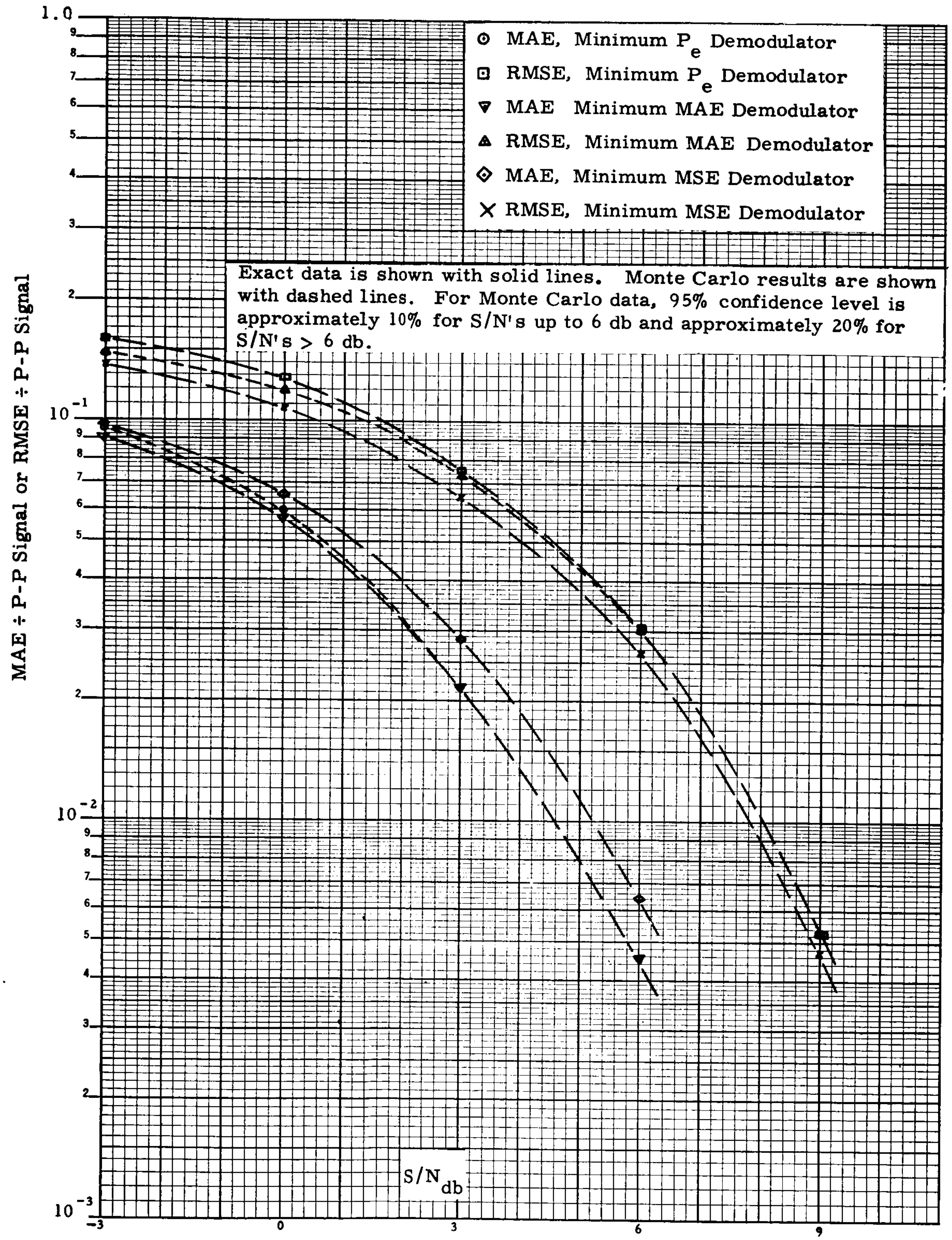

Figure 4-7. Comparison of the Optimal Demodulators, n = 2, m = 6, ρ = .98.

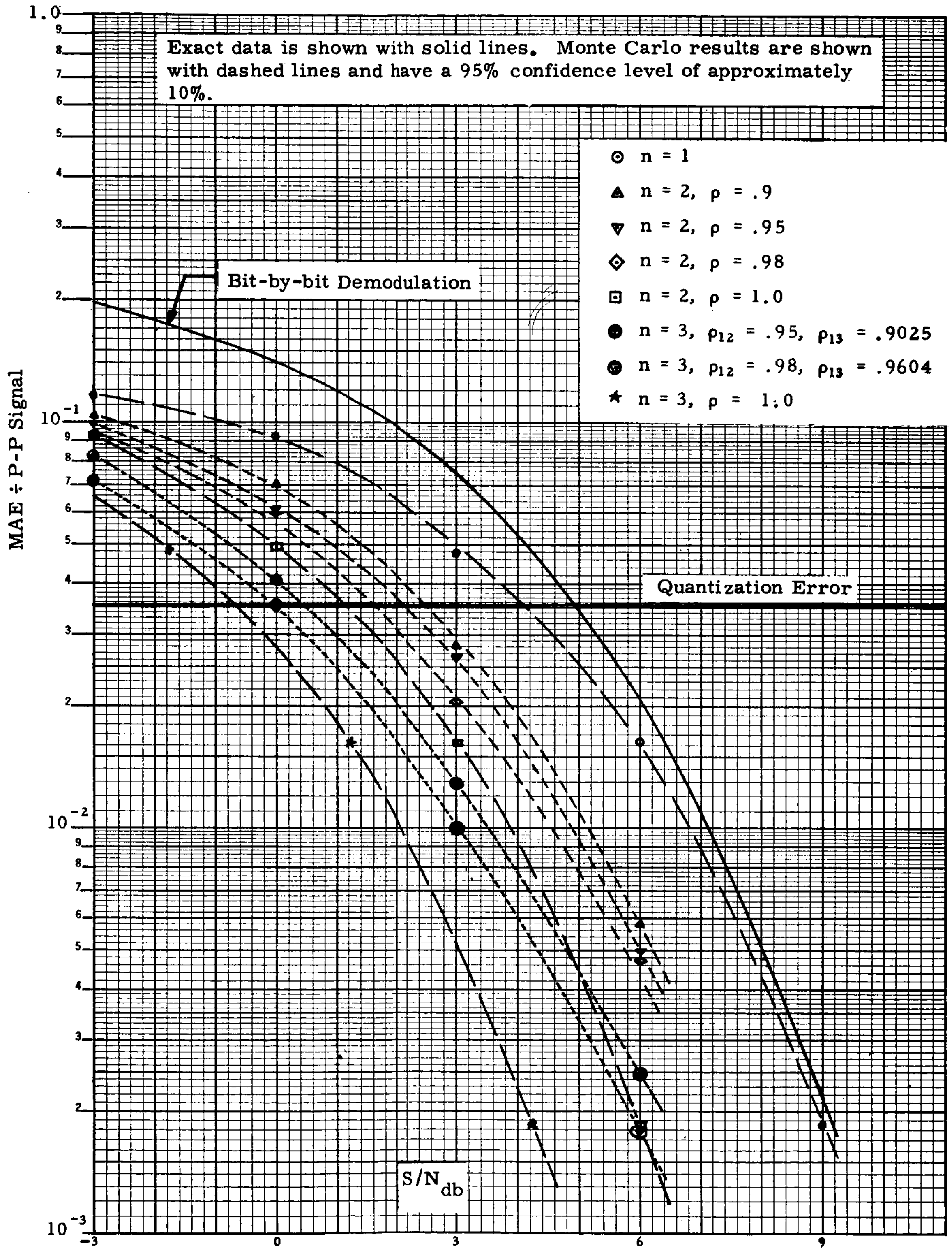

Figure 4-8. MAE for the Minimum MAE Demodulator, m = 3.

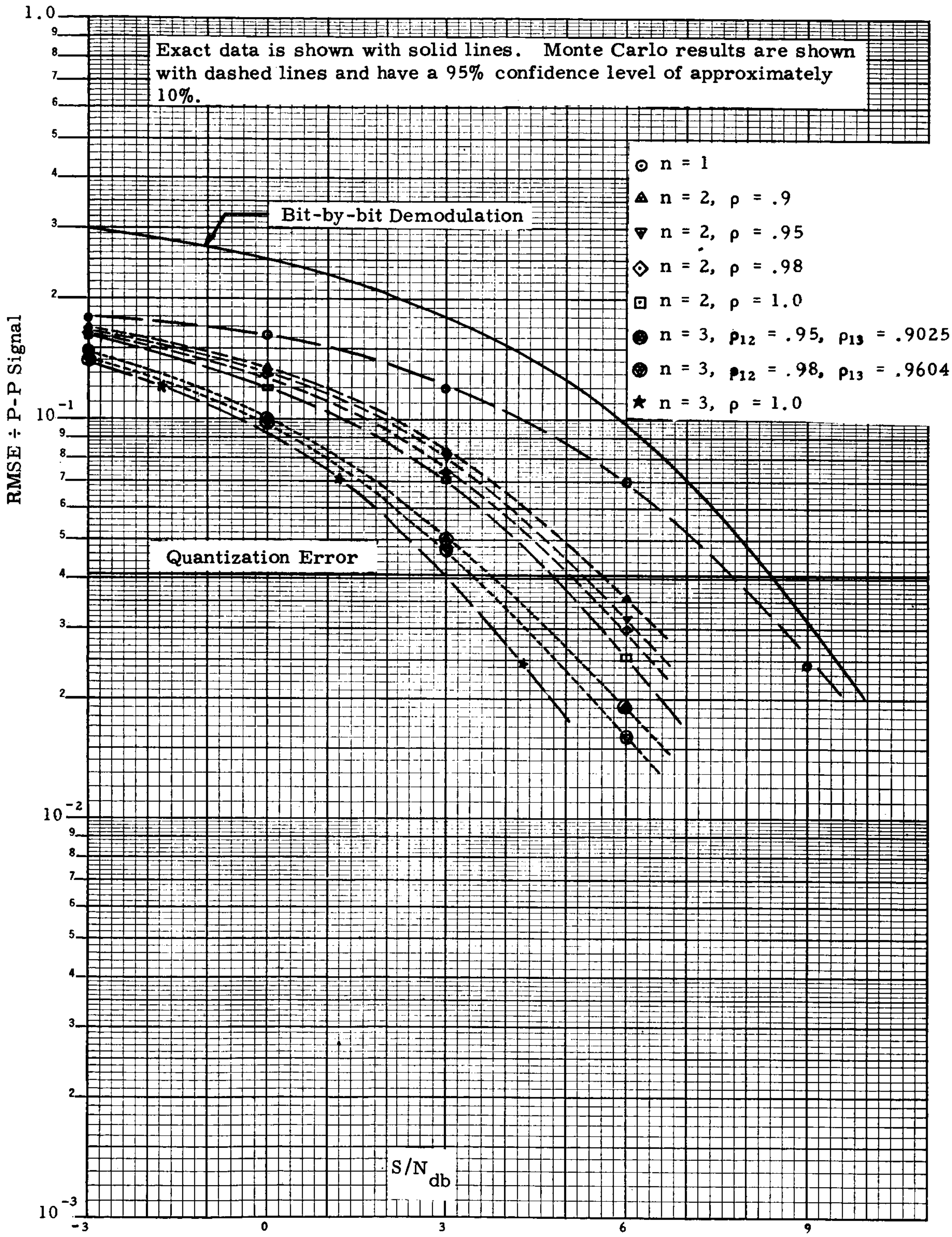

Figure 4-9. RMSE for the Minimum MSE Demodulator, m = 3.

designer can use the performance results of the optimal demodulators in designing the system he needs. For example, let us fix the maximum transmitter power, the noise power, and the word time, T_W. If we go from 6-bit words to 3-bit words, the S/N is increased by 3 db since each bit is transmitted twice as long with 3-bit words. Let us further assume that the sampling rates are such that the expected correlation coefficient between adjacent data samples will be .95. For a comparison of the optimal 6-bit 2-word system with the optimal 2 and 3-word 3-bit systems figure 4-10 results, where the abscissa is the 6-bit S/N in db and the values are normalized by 2^m so that the RMSE comparison is exact. The 6-bit system S/N needed for a demodulation RMSE equal to the quantization error of the 3-bit system is 5.3 db, while the two-word optimal 3-bit system needs a (6-bit) S/N of 2.7 db, and the three-word 3-bit optimal system, 1.2 db. So, as long as the RMSE is specified at or above the 3-bit quantization error, the 3-bit system requires less transmitter power to obtain this RMSE. Of course, if the RMSE is specified lower than the 3-bit quantization error, then a larger number of bits per word must be used. The system designer would also compare 4 and 5-bit systems with the 6-bit system if the RMSE is specified between the 3-bit and 6-bit quantization errors.

Any discussion of the performance of the optimal demodulators would be incomplete without reference to their bias and robustness. These properties of the three optimal demodulators were estimated with a Monte Carlo simulation for the 2 word case. Trends to other values of n should be obvious.

For all of the three optimal demodulators, the expected value of $\hat{Y}_2$ over all possible values of Y_2 is equal to the mean value of Y_2, so the demodulators are unbiased. However, for each particular value of Y_2, there is a bias in the demodulators which will be a function of the signal-to-noise ratio and the correlation coefficient between words.

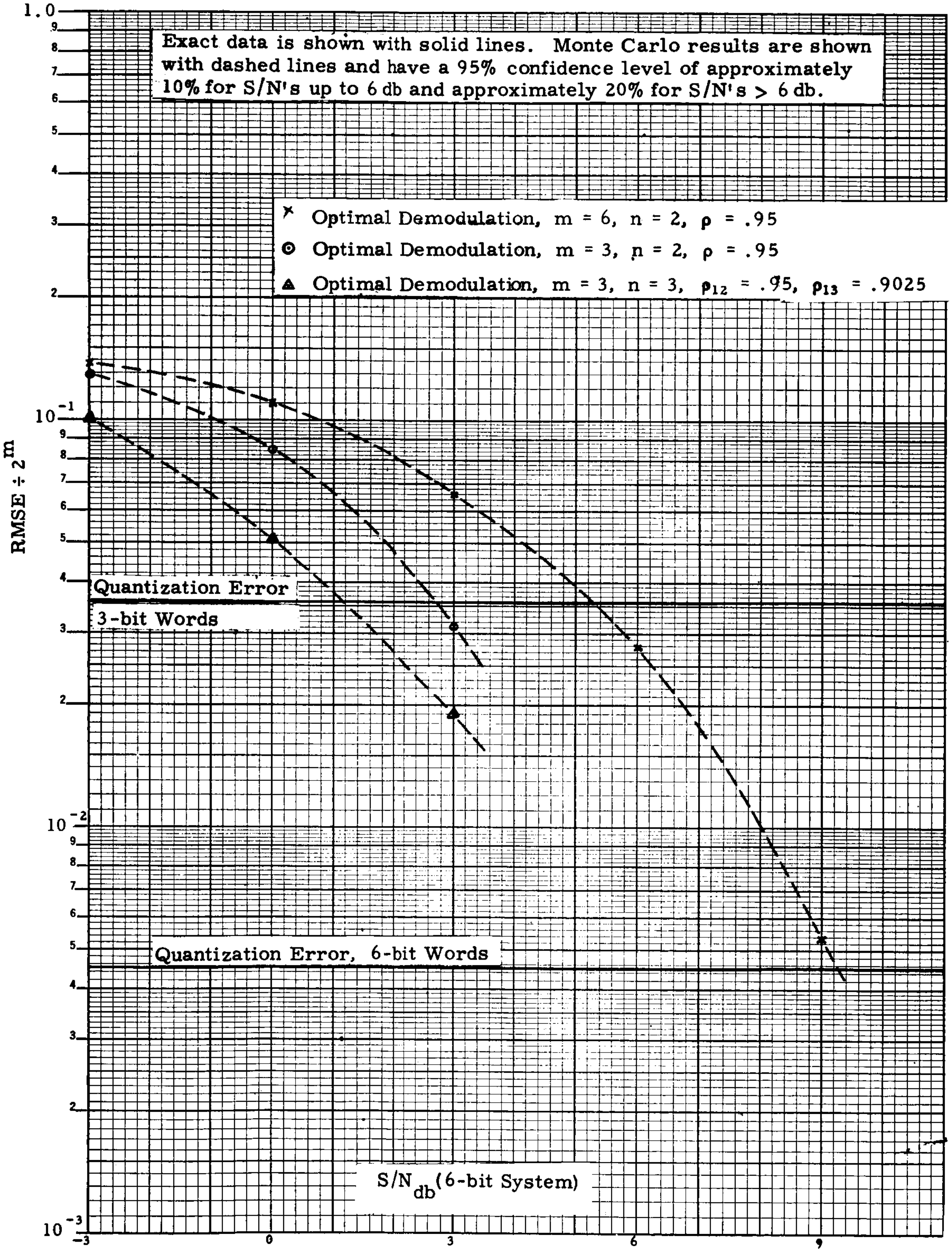

Figure 4-10. System Design Comparison.

To accomplish the Monte Carlo simulation which determines this bias, the computer program for the optimal demodulator simulation (Table A-3-2, Appendix III) was modified so that the desired value of Y_2 is read in and Y_1 is selected from the proper conditional distribution based on Y_2. $\hat{Y}_2$ is then found for each of the three optimal demodulators and averaged over N iterations. The program was additionally modified to give an estimate of the variance of the estimate of the average $\hat{Y}_2$. Since the demodulators are symmetrical about the mean value of Y_2, no values of Y_2 higher than m_Y were used in the simulation. Plots for values greater than m_Y are obtained by plotting $2^m - 1 - \hat{E}(\hat{Y}_2)$ opposite $2^m - 1 - Y_2$. The results of this simulation are given in Tables A-4-2b, A-4-3b, and A-4-4b of Appendix IV for correlation coefficients of .95 and .995 for 6-bit words, and .9 and .95 for 3-bit words.

Figure 4-11 shows this bias for the 6-bit, minimum MAE demodulator, when $\rho = .95$, figure 4-12 for the 6-bit minimum MSE demodulator for $\rho = .95$, and figure 4-13 for the 3-bit minimum MSE demodulator for a ρ of .95. Examination of figures 4-12 and 4-13 again shows that 3-bit demodulator simulations can be used to accurately predict trends in 6-bit and larger optimal demodulators.

There are other types of bias that may be considered. For example, one may be interested in the demodulator bias when the same signal is transmitted each time, i.e., $Y_1 = Y_2 = \cdots = Y_n$. It would be expected that there would be somewhat less bias than that shown in figures 4-11 through 4-13, and the bias in this case could easily be evaluated by means of a Monte Carlo simulation.

To estimate the robustness of the optimal demodulators, the original optimal demodulator program was modified to read in two values each of the S/N and ρ, an actual S/N and ρ to be used in generating the received waveforms, z_{ir}, and a demodulator value of S/N and ρ to be used in the demodulation to get $\hat{Y}_2$. The program had to be further

Monte Carlo results are shown with dashed lines and have a 95% confidence level of < 3.5.

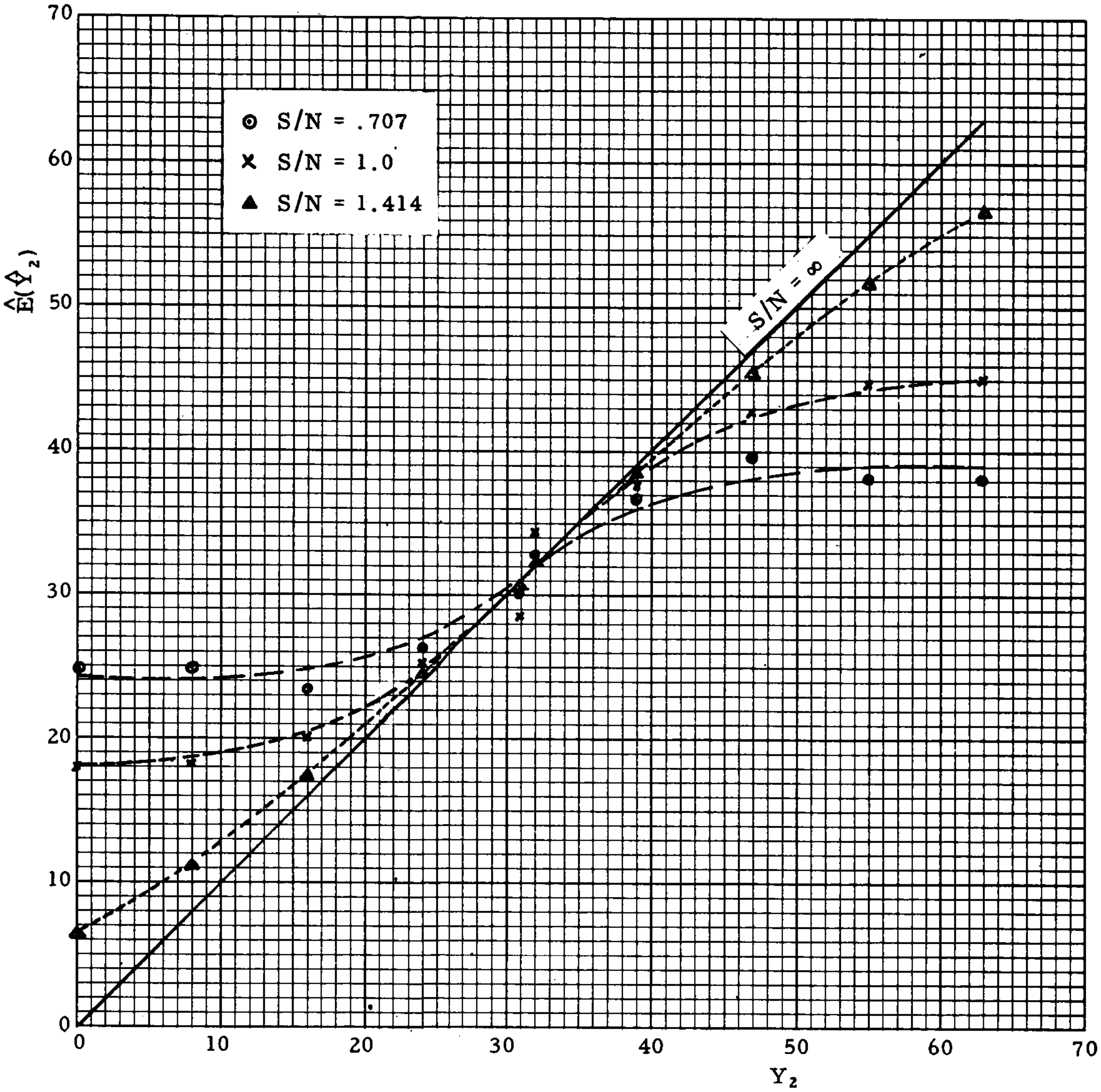

Figure 4-11. Bias, Minimum MAE Demodulator, n = 2, m = 6, ρ = .95.

Monte Carlo results are shown with dashed lines and have a 95% confidence level of < 3.5.

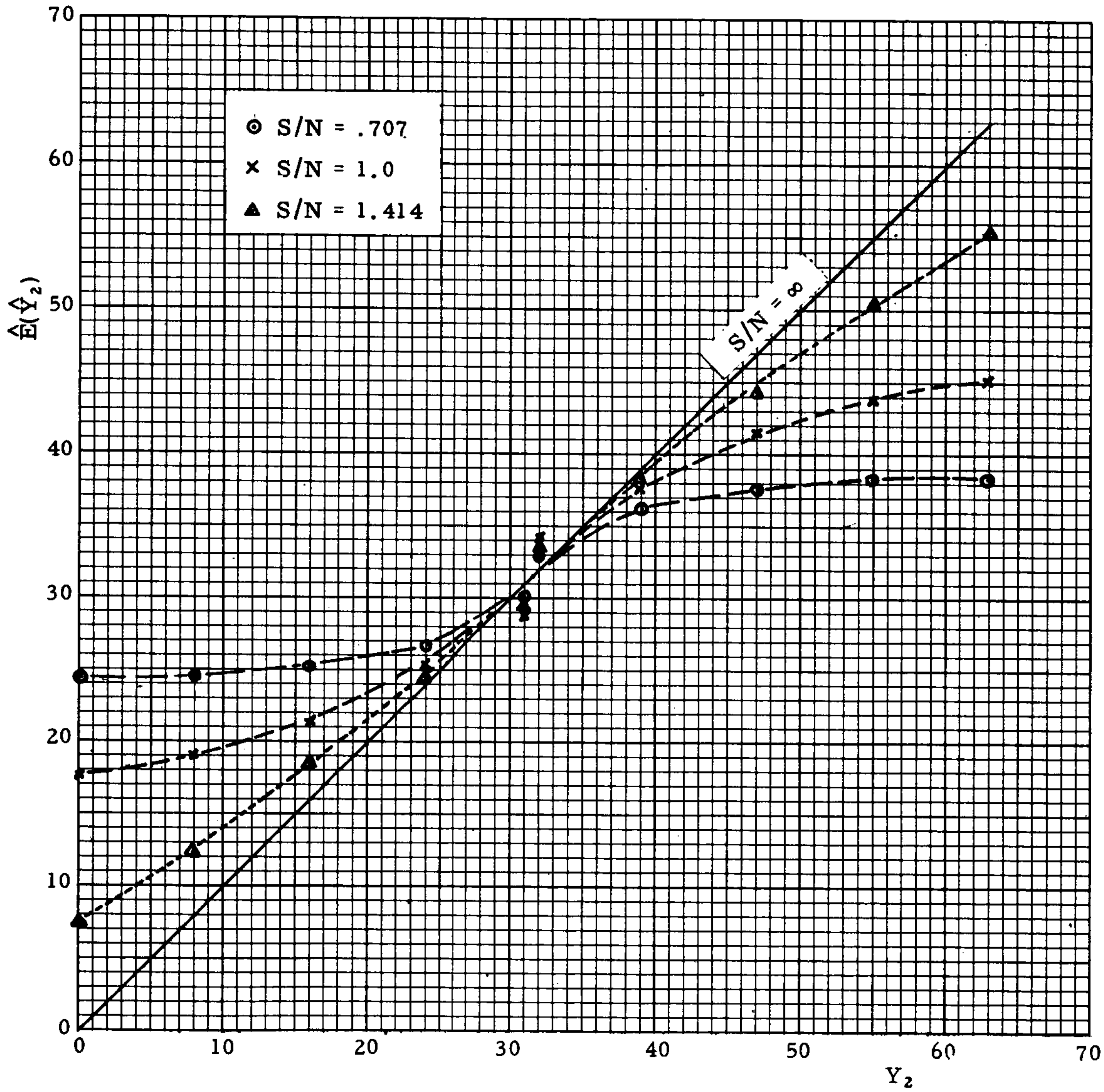

Figure 4-12. Bias, Minimum MSE Demodulator, n = 2, m = 6, ρ = .95.

Monte Carlo results are shown with dashed lines and have a 95% confidence level of < .18.

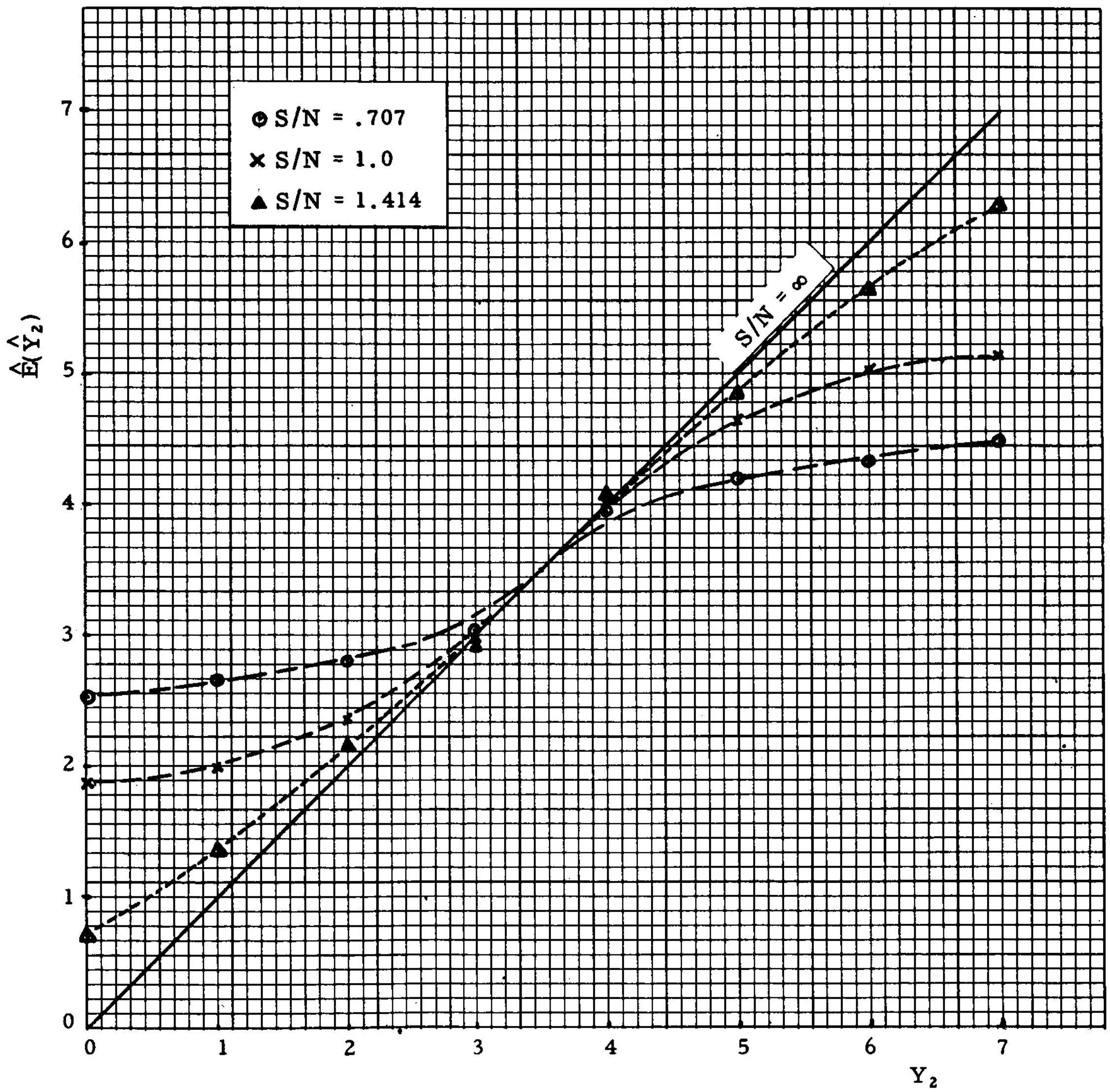

Figure 4-13. Bias, Minimum MSE Demodulator, n = 2, m = 3, ρ = .95.

modified to use equations 2-13a and 2-16a since the computer values of $f_{Y_j|Z}(k|z(t))$ are no longer the true values for use in equations 2-13b and 2-16b. Since the accuracy was decreased by this last change, only 3-bit data was run to estimate the robustness about the $\rho = .95$ line of optimal demodulation. The results for the three optimal demodulators are given in Tables A-4-2c, A-4-3c, and A-4-4c of Appendix IV. The $M\hat{A}E$ robustness of the minimum MAE demodulator and the $RM\hat{S}E$ robustness of the minimum MSE demodulator are shown on figure 4-14, referenced to the optimal demodulator curves from figures 4-8 and 4-9 for $n = 2$, $\rho = .95$. As can be seen from this figure, the optimal demodulators are quite robust even though the S/N may be off as much as 3 db and the value of ρ may be off as far as .9 or .98.

Equations 1-1, 1-2, 1-3 and 1-13 which determine the three optimal demodulators show that these demodulators would be quite complex and expensive to build. Also, due to the large number of operations in equation 1-13, on-line demodulation will be impossible until much faster computer systems are available, particularly for values of n greater than 1. Several suboptimal demodulators that take advantage of the high correlation between data samples, yet are faster and less expensive to build than the optimal demodulators, are discussed in the next chapter.

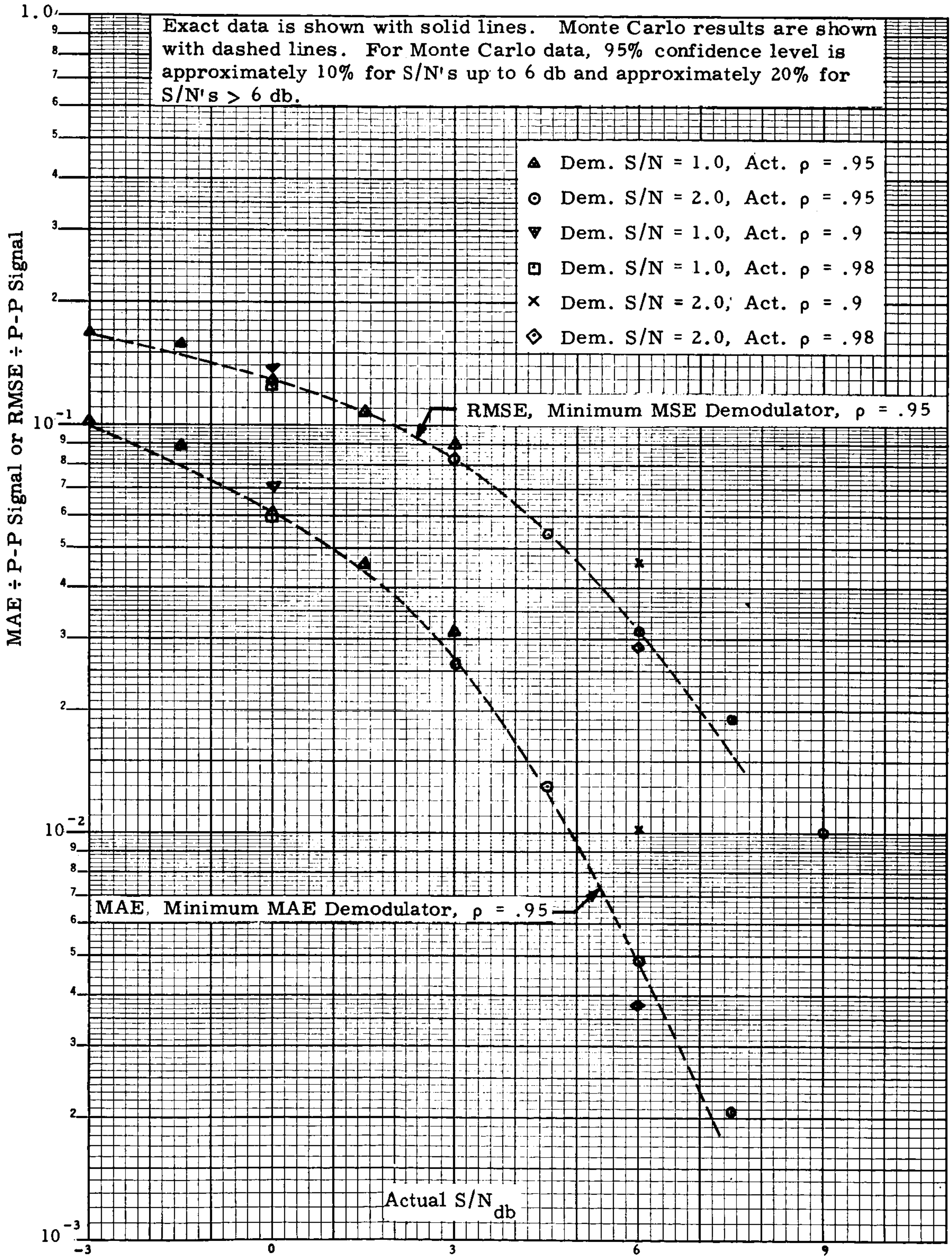

Figure 4-14: Robustness of the Optimal Demodulators, $n = 2$, $m = 3$, Dem. $\rho = .95$.

CHAPTER 5

SUBOPTIMAL PCM DEMODULATION

Considerable improvement in binary PCM demodulation is possible by the use of the optimal demodulators discussed in the first four chapters, and this improvement increases with the number of words (n) considered at one time in the demodulation process. However, as pointed out in Chapter 4, the optimal demodulators are very complex and expensive to build. For the practical case of 6-bit words, the storage of $f_Y(Y_1, \ldots, Y_n)$ requires space for 64^n numbers. For two words this is 4096, for 3 words, 262, 144. Without storage of f_Y, the number of calculations required for the optimal demodulators becomes prohibitively large. Consequently, the optimal demodulators are impractical for $n > 2$, and are practical for $n = 2$ only when a large digital computer is available.

It should be pointed out here that this impracticability of the optimal demodulators does not make the results of the first four chapters useless. The optimal demodulator results give us the limit of possible improvement of binary PCM demodulation when n words, with specified correlation, are considered in the demodulation process and thus are the yardsticks by which we measure future improvements. The optimal demodulators also give some insight into suboptimal demodulation schemes, as will be seen in this chapter.

The results of the optimal demodulators show that taking advantage of the statistical dependence between nearby PCM words will give improved demodulation when the correlation between these words is large. Since the optimal demodulators are rather impracticable, it is natural to look for some simple suboptimal schemes of using this high correlation between nearby words to improve the demodulation process.

Three such schemes are considered and analyzed below.

Smith [S2] suggested the use of $f_{Y_j|z_jY}(k\,|\,z_j(t),\ Y_1, \ldots, Y_{j-1})$, the conditional probability of the quantized value, Y_j, given $z_j(t)$ and $Y_1, \ldots, Y_{j-1}$, in place of $f_{Y_j|z}(k\,|\,z(t))$, where $\hat{Y}_i$ would be used for Y_i since Y_i is not available in the demodulator. This scheme will be referred to as "Smith's Suboptimal Demodulator" in all that follows. For this demodulator

$$f_{Y_j|z_jY}(k\,|\,z_j(t),\ \hat{Y}_i, \ldots, \hat{Y}_{j-1}) = \frac{f_{YZ}(\hat{Y}_1, \ldots, \hat{Y}_{j-1}, k,\ z_j(t))}{f_{zY}(z_j(t),\ \hat{Y}_1, \ldots, \hat{Y}_{j-1})}$$

$$= \frac{f_{z_j|Y}(z_j(t)\,|\,\hat{Y}_1, \ldots, \hat{Y}_{j-1}, k)f_{Y_j|Y}(k\,|\,\hat{Y}_1, \ldots, \hat{Y}_{j-1})f_Y(\hat{Y}_1, \ldots, \hat{Y}_{j-1})}{f_{z_j|Y}(z_j(t)\,|\,\hat{Y}_1, \ldots, \hat{Y}_{j-1})f_Y(\hat{Y}_1, \ldots, \hat{Y}_{j-1})}$$

$$= \frac{f_{z_j|Y}(z_j(t)\,|\,k)f_{Y_j|Y}(k\,|\,\hat{Y}_1, \ldots, \hat{Y}_{j-1})}{f_{z_j}(z_j(t))}$$

since $z_j(t)$ is not a function of $\hat{Y}_1, \ldots, \hat{Y}_{j-1}$. By the arguments of Chapter 1, equation 1-13 then reduces to

$$f_{Y_j|z_jY}(k\,|\,z_j(t), \hat{Y}_1, \ldots, \hat{Y}_{j-1}) = K_4(z)\exp\left\{\frac{1}{\sigma_n^2}\sum_{r=1}^{m}\int_{T_B} z_{jr}y_{j(k)r}\,dt\right\} \cdot f_{Y_j|Y}(k\,|\,\hat{Y}_1, \ldots, \hat{Y}_{j-1}) \qquad (5\text{-}1)$$

This is equivalent to the one-word optimal demodulator, with $f_{Y_j|Y}(k\,|\,\hat{Y}_1, \ldots, \hat{Y}_{j-1})$ substituted for $f_Y(k)$. The values of $f_{Y_j|Y}$ could be precalculated and stored, but the same storage problems that were discussed in the first paragraph of this chapter exist for $j > 2$. A better method would be to calculate each value of $f_{Y_j|Y}$ as it is needed, since, as discussed in Chapter 3, it is given approximately by

$$f_{Y_j|Y}(k|\hat{Y}_1, \ldots, \hat{Y}_{j-1}) \cong \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{-\frac{(k-m)^2}{2\sigma^2}\right\}$$

where σ is a function of σ_Y and the correlation coefficients between the data samples, and m is a function of m_Y, the ρ's, and $\hat{Y}_1, \ldots, \hat{Y}_{j-1}$. Equation 5-1 then becomes

$$f_{Y_j|z_j Y}(k|z_j(t), \hat{Y}_1, \ldots, \hat{Y}_{j-1}) \cong K_5(z) \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} z_{jr} y_{j(k)r} \cdot dt - \frac{(k-m)^2}{2\sigma^2}\right\} \tag{5-2}$$

Equation 5-2 and the expressions for m and σ, along with equations 1-1, 1-2, and 1-3, then specify the three suboptimal demodulators proposed by Smith.

The standard deviation, σ, can be precalculated and stored based on the ρ's. The only effect then on Smith's suboptimal demodulator of an increase in the number (j) of words considered is in the calculation of m. Consequently, as j increases, the complexity of this suboptimal demodulator changes very little.

Since $\hat{Y}_j$ is different for the three versions of this demodulator, only one was analyzed, Smith's suboptimal demodulator corresponding to the minimum MSE demodulator. A Monte Carlo simulation of this demodulator for the two word, 3-bit word case was run. For this case equation 5-2 becomes

$$f_{Y_2|z_2 Y_1}(k|z_2(t), \hat{Y}_1) = K_5(z) \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{3} \int_{T_B} z_{2r} y_{2(k)r} dt - \frac{(k-m)^2}{2\sigma^2}\right\} \tag{5-3}$$

and

$$m = m_Y(1-\rho) + \rho\hat{Y}_1$$

$$\sigma = \sigma_Y \sqrt{1-\rho^2} \tag{5-4}$$

Equation 1-3 is rewritten as

$$\hat{Y}_2 = \sum_{k=0}^{7} k\, f_{Y_2|z_2 Y_1}(k|z_2(t), \hat{Y}_1) \tag{5-5}$$

In the Monte Carlo simulation, $\hat{Y}_1$ is not available until one demodulation has been accomplished, and to insure that the demodulation process has progressed far enough for initial effects to be smoothed out, four successive demodulations were simulated for each iteration. The estimators used were by necessity those specified by equations 2-13a and 2-16a. The computer program for Smith's suboptimal demodulator is given in Table A-3-3 of Appendix III.

Two values of ρ were run, .9 and .95. The results are given in Table A-4-5 of Appendix IV and are plotted on figure 5-1. As can be seen from this figure, the results for $\rho = .9$ were better than for $\rho = .95$, particularly at the higher S/N's. Some test runs at ρ's higher than .95 and lower than .9 were run to substantiate this, and it was found that the performance did deteriorate for high ρ, but that for ρ's smaller than .9 the trend reversed and the performance also got worse. Some insight into this phenomenon can be gained by considering the combined form of equations 5-3 and 5-4,

$$f_{Y_2|z_2 Y_1}(k|z_2, \hat{Y}_1) = K_5(z) \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{3} \int_{T_B} z_{2r} y_2(k)_r dt - \frac{(k - m_Y(1-\rho) - \rho\hat{Y}_1)^2}{2\sigma_Y^2(1-\rho^2)}\right\} \tag{5-6}$$

When ρ is close to 1, the term $1 - \rho^2$ in the denominator of the negative part of the exponent tends to drive the exponential to 0 unless k is close to $\hat{Y}_1$, thus magnifying any errors in previous demodulations. For low ρ, the first term in the exponent becomes dominant, and the use of $\hat{Y}_1$ has little effect on $\hat{Y}_2$.

Based on the unpromising results on figure 5-1, no further

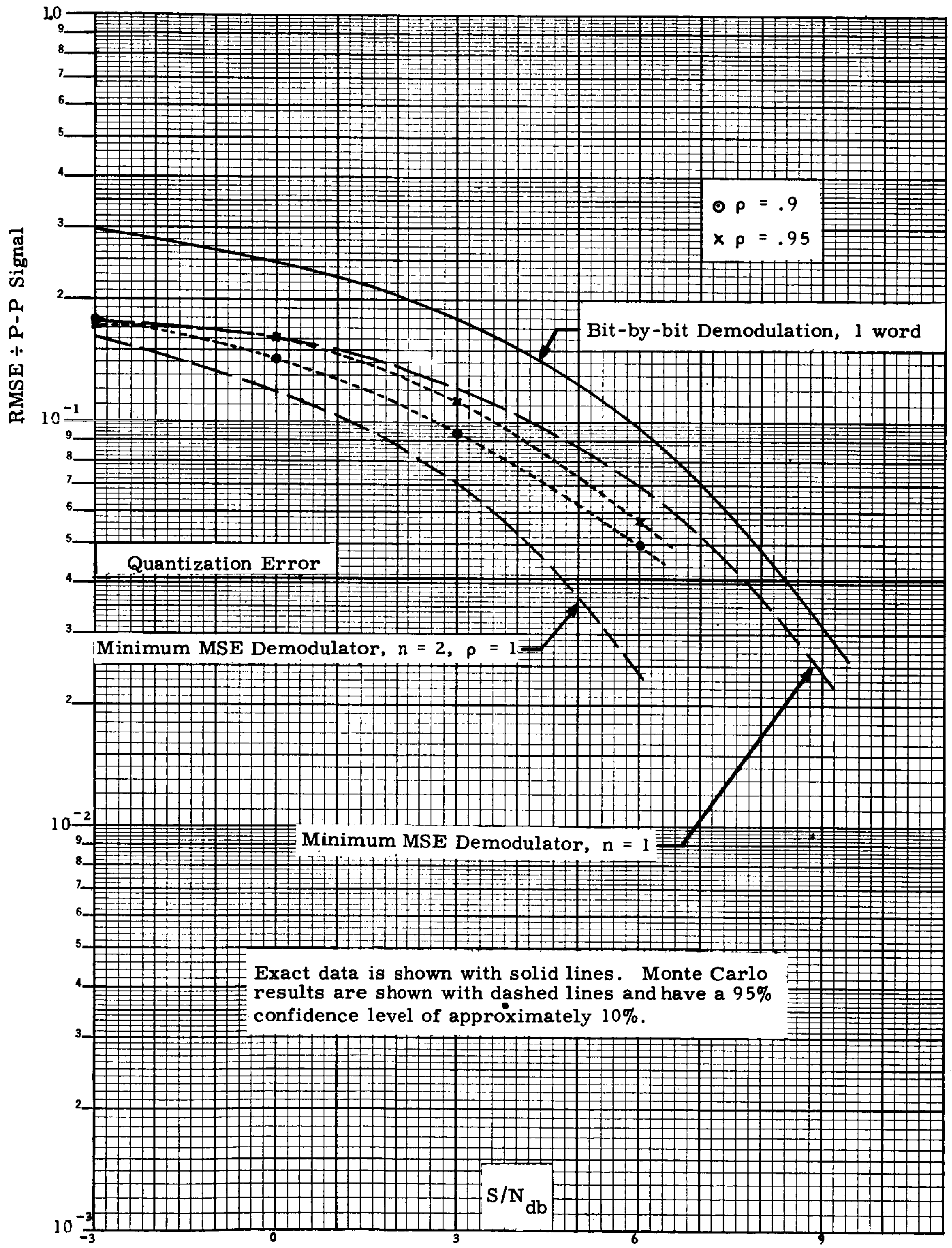

Figure 5-1. RMSE, Smith's Suboptimal Demodulator, m = 3, n = 2.

simulation of Smith's suboptimal demodulator was run. Considering more words in each demodulation would give some improvement, but it is doubtful that the improvement would be significant.

The second suboptimal demodulator considered incorporates a demodulation scheme suggested by Professor L. L. Rauch. In "Rauch's suboptimal demodulator" a weighted average over n words of individual bits is correlated with the reference signals $g_1(t)$ and $g_2(t)$ ($f_1(t)$ and $f_2(t)$ in the general case) to determine the estimate of that bit. In equation form

$$
\begin{aligned}
G_{1r} &= \int_{T_B} g_1(t)\,(w_{1r} z_{1r} + \cdots + w_{jr} z_{jr} + \cdots + w_{nr} z_{nr})\,dt \\
G_{2r} &= \int_{T_B} g_2(t)\,(w_{1r} z_{1r} + \cdots + w_{jr} z_{jr} + \cdots + w_{nr} z_{nr})\,dt
\end{aligned}
\tag{5-7}
$$

$$
\hat{y}_{jr} = \begin{cases} g_1(t) & \text{if } G_{1r} > G_{2r} \\ g_2(t) & \text{if } G_{2r} > G_{1r} \end{cases}
\tag{5-8}
$$

where r corresponds to the r^{th} bit. The demodulator always estimates the r^{th} bit of the center word of the sequence of n words, i.e., $n = 2j - 1$, and the weightings must sum to 1, i.e., $\sum_{i=1}^{n} w_{ir} = 1$, $r = 1, \ldots, m$.

Although the main advantage of this demodulator is its simplicity, even for large n, the performance evaluation of this demodulator by a Monte Carlo simulation proved unwieldly. The large number of Gaussian random variables that must be generated for large n requires considerable computer time, and the simulation must first determine the best set of weightings, w_{ir}, by trial and error before the performance can be evaluated. Here the direct calculation of the optimal weightings and then the MAE and RMSE appears to be the best approach.

The calculations were made for five words ($n = 5$), with the third word being estimated ($j = 3$). Words number 2 and 4 were assumed to have a correlation coefficient of ρ_{12} with the third word. Words 1 and 5 were assumed to have a correlation coefficient of ρ_{13} with the third word, and a correlation coefficient of ρ_{12} with the 2nd and 4th words respectively. For this case

$$\begin{aligned} f_Y(Y_1, Y_2, Y_3, Y_4, Y_5) &= f_Y(Y_1, Y_2 | Y_3, Y_4, Y_5)\, f_Y(Y_3, Y_4; Y_5) \\ &= f_Y(Y_1, Y_2, | Y_3)\, f_Y(Y_3, Y_4, Y_5) \\ &= \frac{f_Y(Y_1, Y_2, Y_3)\, f_Y(Y_3, Y_4, Y_5)}{f_Y(Y_3)} \end{aligned}$$

The correlation coefficients between words 1 and 4 (and words 2 and 5), ρ_{14}, and between words 1 and 5, ρ_{15}, are fixed by the above assumptions and could be determined by the same methods which were used in Chapter 3 for the three word case. The computer program first computes the probability that a certain sequence of signal bits was transmitted in the r^{th} bit position, denoted by $PBS_{\ell r}$, $\ell = 1, \ldots, 2^5$. These values are used to compute the optimal weightings[1], w_{ir}, and the probability that r^{th} bit is in error, denoted by PBE_r. Using the PBE_r's, the probability of an error of a magnitude of k is calculated (denoted by P_k) and this is used to compute the MAE and RMSE. The computer program to do this is discussed in Appendix III and is given in Table A-3-4 of that appendix.

The computation was done for 3-bit words, and for three sets of correlation coefficients ranging from very high to medium. For six bit words, computation of $PBS_{\ell r}$ requires an excessive amount of computer time, as can be seen by considering the computer program in Appendix III. The computation of $PBS_{\ell r}$ requires the computation of $f_Y(Y_1, Y_2, Y_3)$,

[1] The computer program (Appendix III) was set up to select the set of weightings for a particular bit that minimizes the probability of an error in that bit. This will not in general give the set of weightings that will minimize either the MAE or the MSE.

which is too large to store for m = 6, and therefore, must be recomputed each time it is used. Also, in moving from 3 to 6-bit words, the number of operations to get each $PBS_{\ell r}$ goes from 4^5 to 32^5, resulting in too large an increase in required computer time. However, it should be noted that the bit-error probabilities for the 3 bits in 3-bit words are the same as the bit-error probabilities for the first three bits of 6-bit words since the PBS's are the same. To get some idea of the 6-bit word performance of Rauch's suboptimal demodulator, bit-error probabilities were assumed for the 4th, 5th, and 6th bit. PBE_6 was taken to be the same as with no averaging (i.e., $w_{3r} = 1$), and PBE_4 and PBE_5 were interpolated linearly between PBE_3 and PBE_6. Using these assumed bit-error probabilities, the MAE and RMSE were calculated for 6-bit data for the same three sets of correlation coefficients as were used for the 3-bit data. Also, the robustness of Rauch's demodulator was determined for 3-bit words by running the demodulator at the weightings for other S/N's and ρ's than the values set in the program.

The computer results for Rauch's suboptimal demodulator are given in Table A-4-6 of Appendix IV. These results are also shown on figures 5-2 and 5-3 (3-bit), figure 5-4 (6-bit) and figure 5-5 (robustness). For reference, the $\rho_{12} = \rho_{13} = 1.0$ line is shown 7 db to the left of the one-word, bit-by-bit demodulator line, and the results of the minimum MAE or MSE demodulator for $\rho = 1$, and n = 1, 2, and 3 are shown also. As can be seen from these figures, this suboptimal demodulator gives much improved performance over the present day (one-word, bit-by-bit) demodulator, particularly for high values of ρ_{12} and ρ_{13} and low values of S/N. In fact, for $\rho_{12} = .995$ and $\rho_{13} = .99$ the performance of Rauch's demodulator is close to or better than the performance of the corresponding optimal 2-word demodulator. It should be noted that even for very high correlation coefficients the performance is considerably poorer than for $\rho = 1.0$, but this should be expected for suboptimal demodulation schemes.

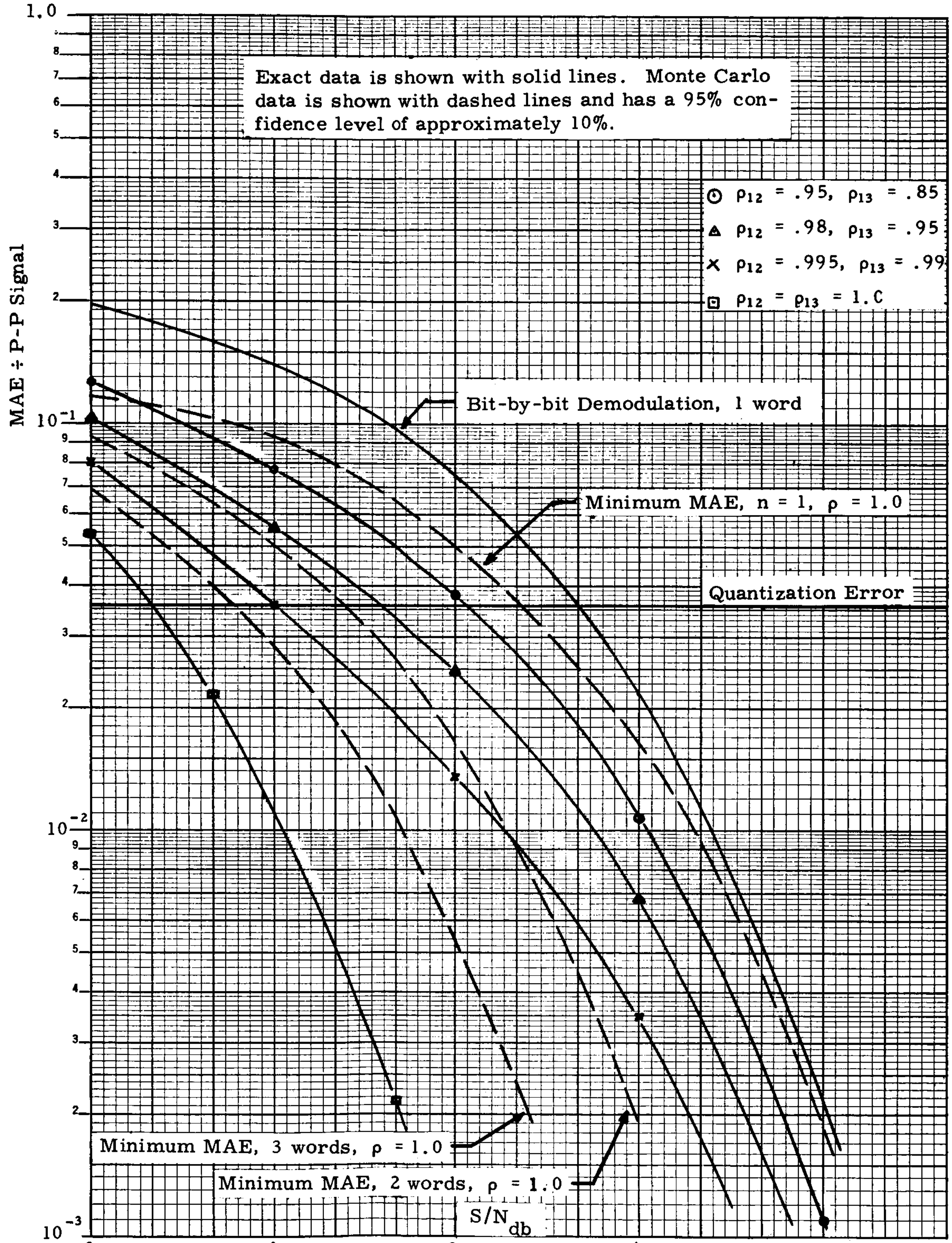

Figure 5-2. MAE, Rauch's Suboptimal Demodulator, n = 5, m = 3.

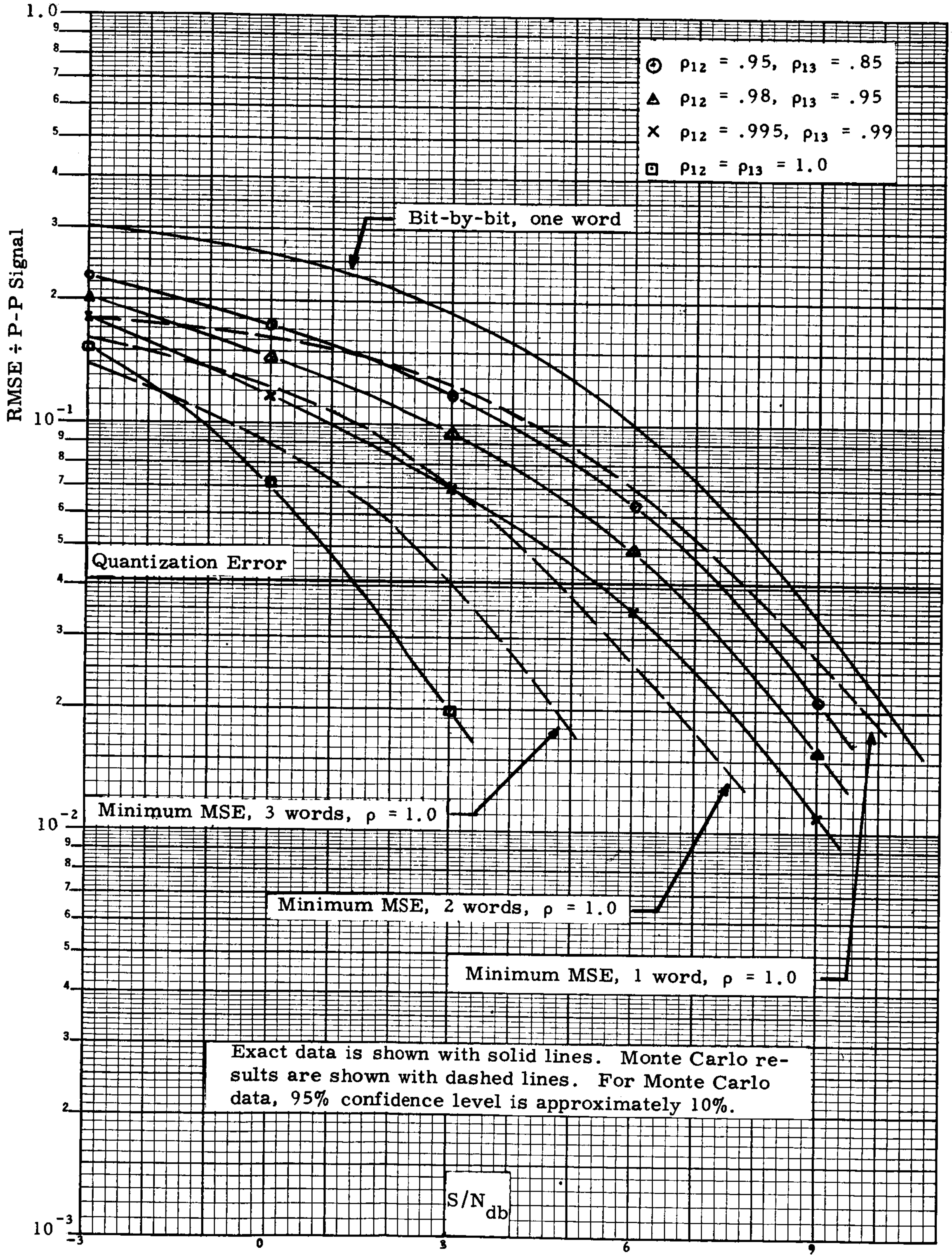

Figure 5-3. RMSE, Rauch's Suboptimal Demodulator, n = 5, m = 3.

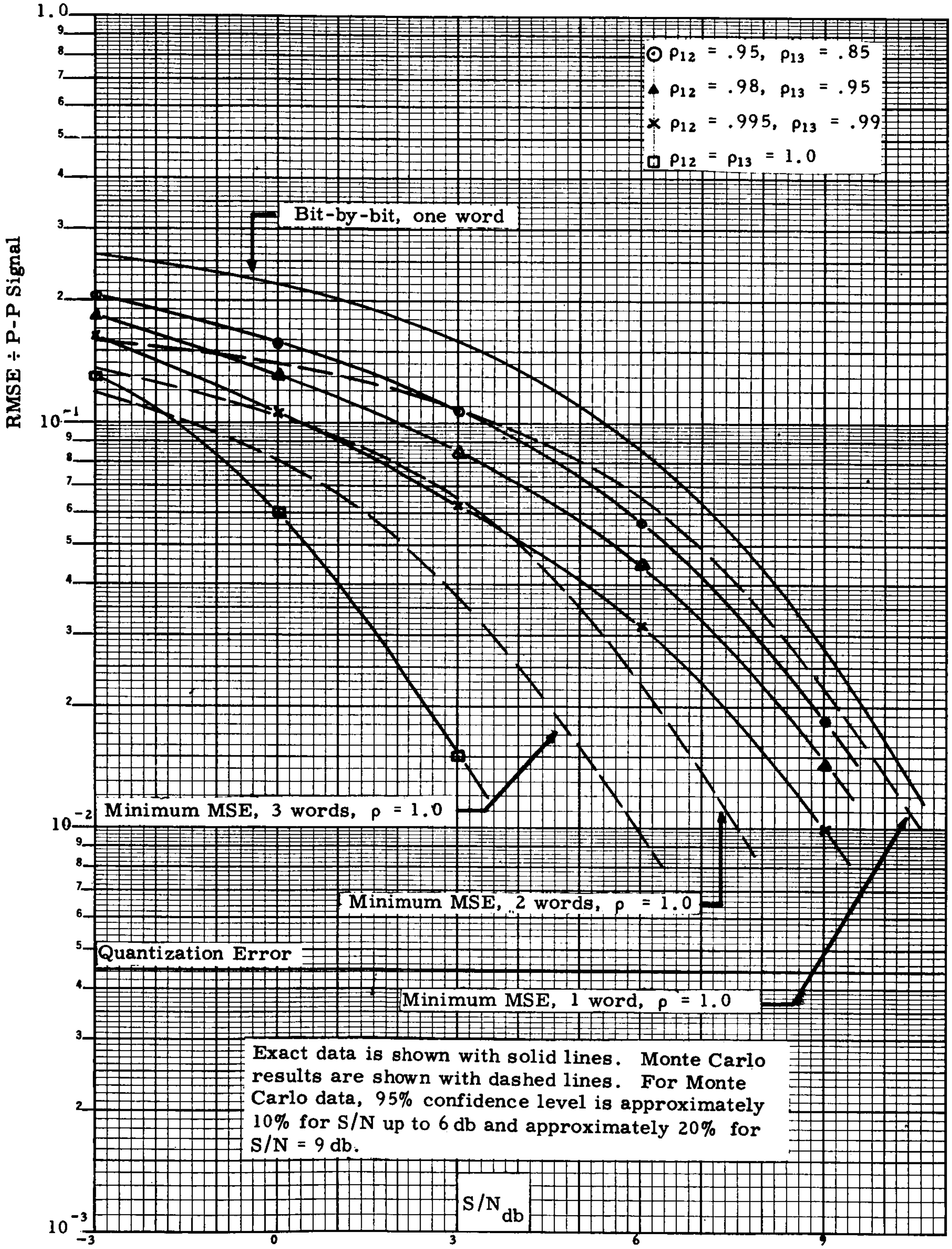

Figure 5-4. RMSE, Rauch's Suboptimal Demodulator, n = 5, m = 6 (with assumed PBE's).

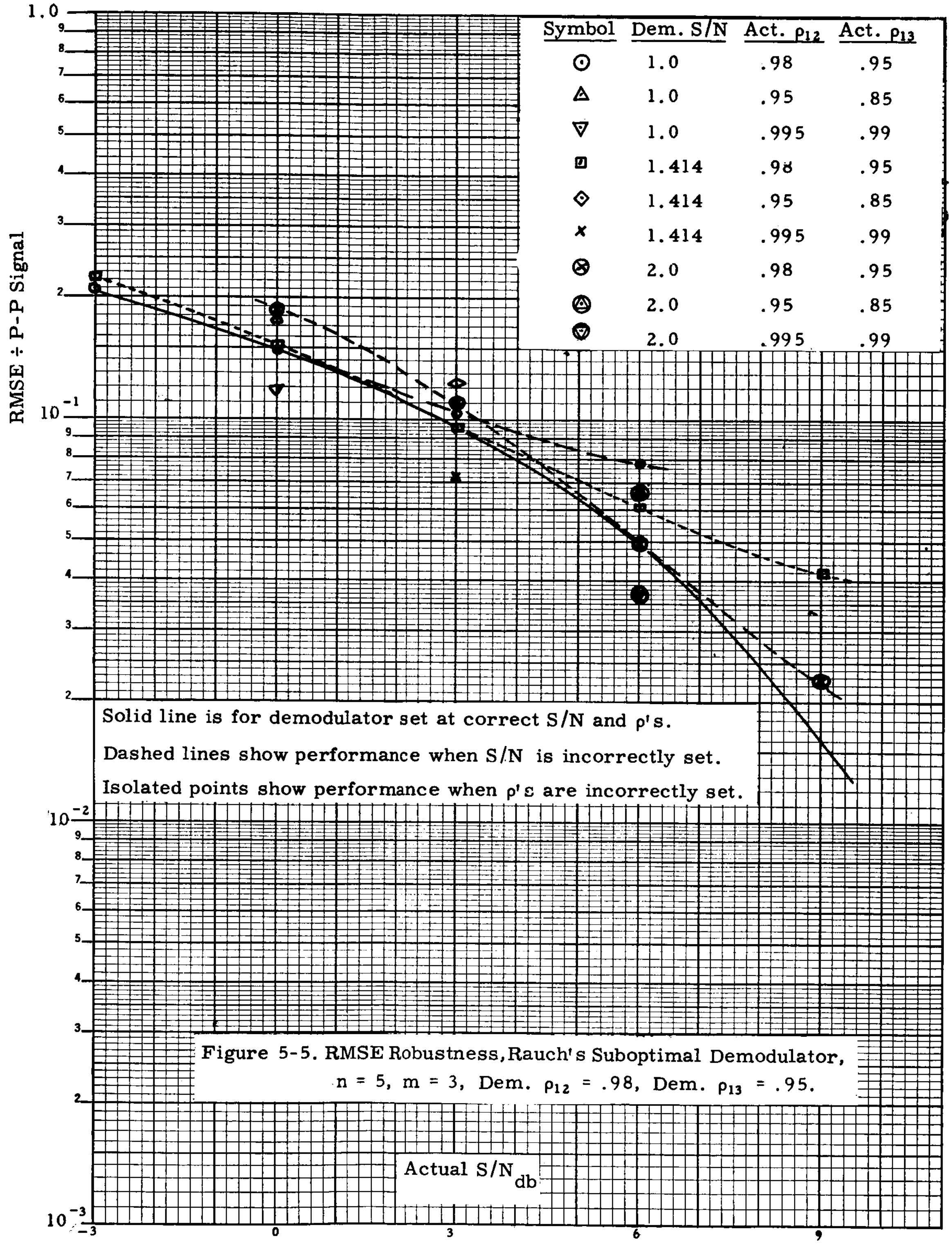

Figure 5-5. RMSE Robustness, Rauch's Suboptimal Demodulator, $n = 5$, $m = 3$, Dem. $\rho_{12} = .98$, Dem. $\rho_{13} = .95$.

A look at the optimal weightings, w_{ir}, in Table A-4-6a of Appendix IV shows that the consideration of more than 5 words in the demodulation process should give more improvement in the bit-error probability, particularly at the lower S/N's, higher ρ's, and for the first or second bits where the weighting on words 1 and 5 is fairly high. Along with this, it should be noted that errors in the first bit (and to a lesser extent, the second and succeeding bits) have the most effect on the absolute error or square error, so decreasing the probability of an error in the leading bits can significantly reduce the MAE and RMSE. Therefore, using more words in each demodulation should give improved demodulator performance.

Figure 5-5 shows that the robustness of this demodulator is quite good, except where the demodulator S/N is more than 3 db lower than the actual S/N. This is to be expected since the demodulator is then weighting words 1, 2, and 4 and 5 heavier than optimal and therefore increasing the chances of an error.

For n = 1, w_{ir} must equal 1, and Rauch's suboptimal demodulator becomes the present day, bit-by-bit demodulator. In Chapter 4, it was seen that the optimal, one-word demodulator gave an improvement in performance over the present day demodulator, particularly at low S/N. It would be expected that a similar weighting by f_Y, which is in essence what the optimal, one-word demodulator does, would improve the performance of Rauch's suboptimal demodulator for n > 1. This reasoning led to the third suboptimal demodulator that will be considered, which can be described as follows. Let

$$H_{jr} = w_{ir}z_{ir} + \cdots + w_{jr}z_{jr} + \cdots + w_{nr}z_{nr} \tag{5-9}$$

and

$$h(k|z(t)) = K_6(z) \exp\left\{\frac{1}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} H_{jr}\, y_{j(k)r}\, dt\right\} f_Y(k) \tag{5-10}$$

Then the counterpart of the minimum MSE demodulator is given by

$$\hat{Y}_j = \sum_{k=1}^{2^m-1} k\, h(k \mid z(t)) \tag{5-11}$$

The use of equations 1-1 and 1-2 with h replacing $f_{Y_j|z}$ would give the counterparts of the minimum P_e and minimum MAE demodulators. However, only the demodulator corresponding to the minimum MSE demodulator (equation 5-11) is analyzed here.

A Monte Carlo simulation was used to evaluate the performance of suboptimal demodulator number three, where the weightings, w_{ir}, had been determined in the evaluation of Rauch's suboptimal demodulator. It was found that better performance could be obtained if the effects of $f_Y(k)$ were weighted in such a way that it would have a smaller effect when the contribution for the other received words was the greatest in H_r (i.e., when the w_{ir} were more nearly equal) and the largest effect when z_{jr} made the largest contribution to H_{jr}. In the actual simulation this was accomplished by weighting H, and equation 5-10 becomes

$$h(k \mid z(t)) = K_6(z) \exp\left\{\frac{WT}{\sigma_n^2} \sum_{r=1}^{m} \int_{T_B} H_{jr}\, y_{(k)r}\, dt\right\} f_Y(k) \tag{5-12}$$

where WT is the weighting factor. The best weighting factor was determined approximately by trial and error Monte Carlo simulations. The weighting factor that gave the least RMSE was not the weighting that gave the least MAE, in fact the two were considerably different with the weighting factor for the least MAE being larger. The weighting factors in Table A-4-7 are those that gave the best RMSE performance, since the version of suboptimal number 3 analyzed was the one corresponding to the minimum MSE demodulator. It should also be noted that the addition of this weighting factor adds one more degree of freedom to the demodulator, thereby increasing the complexity and adding to the robustness problem.

The computer simulation was done for five (n = 5) 3-bit words, with the same assumptions on the correlation coefficients as were used in the analysis of Rauch's suboptimal demodulator. For this case

$$f_{x_1|x_2,x_3}(x_1^*|x_2^*,x_3^*) = \frac{f_X(x_1^*,x_2^*,x_3^*)}{f_X(x_2^*,x_3^*)}$$

and for Gaussian data this becomes

$$f_{x_1|x_2,x_3}(x_1^*|x_2^*,x_3^*) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{-\frac{(x_1^* - m)^2}{2\sigma^2}\right\}$$

where

$$\sigma = \sigma_Y\sqrt{\frac{1 - 2\rho_{12}^2(1 - \rho_{13}) - \rho_{13}^2}{1 - \rho_{12}^2}}$$

$$m = \left(1 - \frac{\rho_{12}(1 - \rho_{13})}{1 - \rho_{12}^2} + \frac{\rho_{12}^2 - \rho_{13}}{1 - \rho_{12}^2}\right) m_Y + \left(\frac{\rho_{12}(1 - \rho_{13})}{1 - \rho_{12}^2}\right) x_2^*$$
$$- \left(\frac{\rho_{12}^2 - \rho_{13}}{1 - \rho_{12}^2}\right) x_3^*$$

x_3^* is selected from a normal (m_Y, σ_Y) distribution. x_2^* and x_4 are then selected from a normal distribution with mean $m_Y(1 - \rho_{12}) + \rho_{12}x_3^*$ and standard deviation $\sigma_Y\sqrt{1 - \rho_{12}^2}$. x_1^* and x_5^* are then selected from a normal distribution with mean m and standard deviation σ as given in the last two equations above (for x_5^*, x_4^* replaces x_2^* in the expression for m). $\hat{Y}_3$ is formed from equations 5-12 and 5-11. The estimate of the MAE and MSE and the estimates of the variances of these estimates are made using equations 2-13a, 2-14, 2-15, 2-16a, 2-17, and 2-18. The computer program for the simulation of this demodulator is given in Table A-3-5 of Appendix III and the results of the simulation for the same sets of ρ_{12}'s and ρ_{13}'s as were used with Rauch's demodulator are given in Table A-4-7 of Appendix IV and are plotted on figure 5-6.

Going from bit-by-bit demodulation as in Rauch's demodulator to word demodulation as in suboptimal number 3 requires a substantial increase in complexity in the demodulator. Rewriting the exponential term in equation 5-12 as

$$\exp\left\{\frac{WT}{\sigma_n^2}\sum_{r=1}^{m} w_{1r}\int_{T_B} z_{1r}\, y_{j(k)r}\,dt + \cdots + w_{nr}\int_{T_B} z_{nr}\, y_{j(k)r}\,dt\right\}$$

it can be seen that suboptimal demodulator number 3 requires two correlators whose outputs are $\int_{T_B} z_{ir} g_1(t)\,dt$ and $\int_{T_B} z_{ir} g_2(t)\,dt$. These values are then fed into a small digital computer where they are stored and used with the previous outputs (already stored in the computer) to form the needed quantities in equation 5-12 (the values of $f_Y(k)$ and the w_{ir}'s would be previously stored in the computer). Then $\hat{Y}$ is formed according to equation 5-11. Although this demodulator is more complex than Rauch's demodulator, like Rauch's demodulator the complexity increases only slightly as n is increased. The user would have to determine if the added performance justifies the added complexity. The robustness characteristics should be similar to those of Rauch's demodulator, with a slight added problem due to the weighting factor.

Figure 5-7 shows a RMSE comparison of the three suboptimal demodulator performances. Smith's demodulator has a comparatively poor performance except at low S/N's. Rauch's demodulator gives good performance, particularly for high ρ's. A considerable amount of added improvement can be had at the price of added complexity over Rauch's demodulator with suboptimal number three.

For one optimization criteria there is only one possible optimal demodulator, but there is no such limit on the number of possible

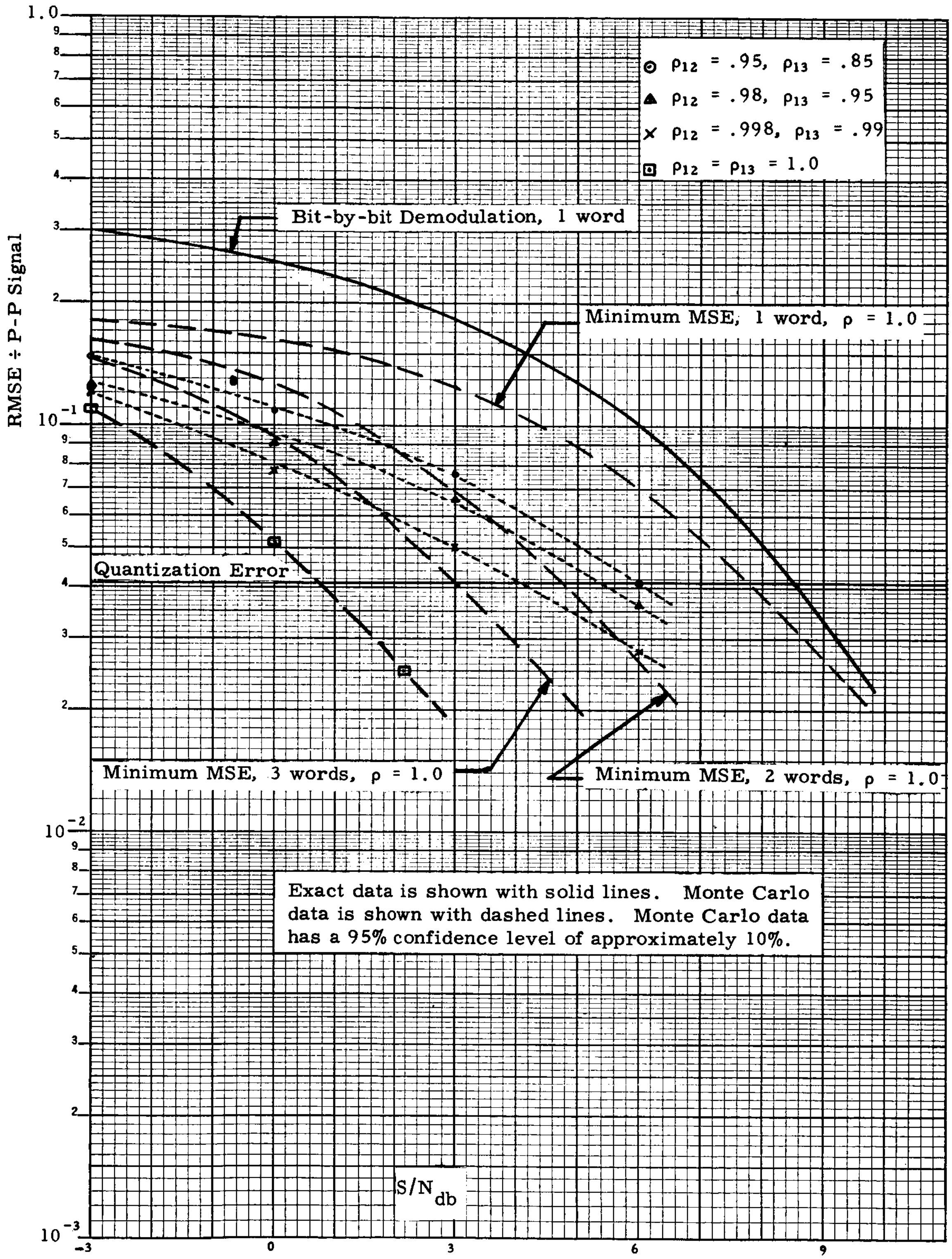

Figure 5-6. RMSE, Suboptimal Demodulator No.3, n = 5, m = 3.

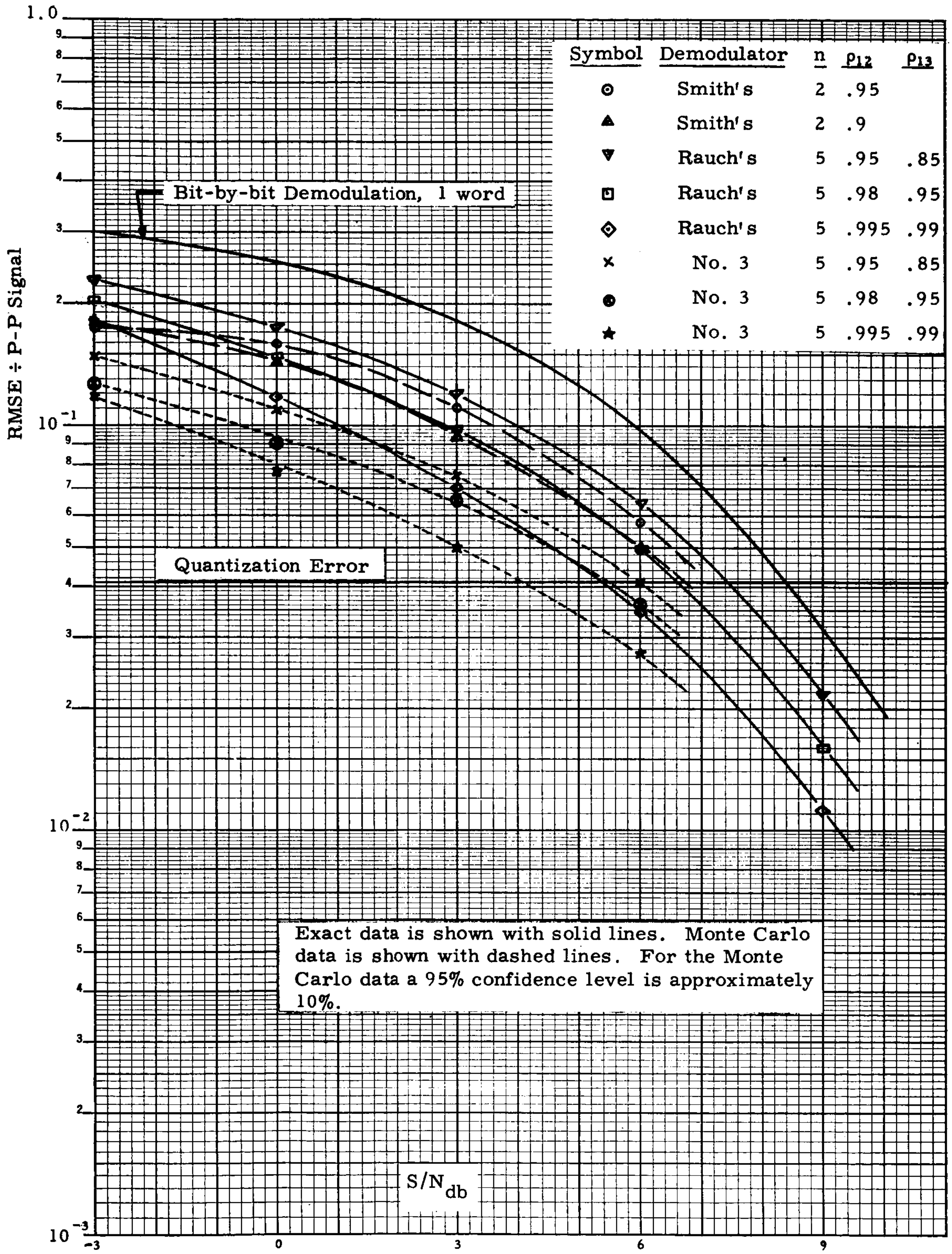

Figure 5-7. RMSE Comparison of the Suboptimal Demodulators, m = 3.

suboptimal demodulators. Therefore, the three suboptimal schemes analyzed above do not exhaust the possible suboptimal demodulators that could be used. However, these suboptimal demodulators do show that high statistical dependence between nearby data samples can be used to significantly improve demodulator performance in a practical way.

CHAPTER 6

CONCLUSIONS AND POSSIBLE EXTENSIONS

In reference S2, Smith showed that high statistical dependence between PCM data samples could be used in the demodulation process to reduce the probability of a word error. In the first four chapters of this dissertation it has been shown that this improvement in the demodulation is more pronounced when more meaningful measures of demodulator performance are considered, and that the demodulators that minimize the mean absolute error or mean square error show the same improvement as the minimum-error-probability demodulator that Smith considered.

Chapter 5 points out that high correlation between data samples can be used suboptimally to significantly improve PCM demodulation. Although the results are not as good as the optimal demodulators for the same number of words considered in each demodulation, the complexity of the suboptimal demodulators will not increase to the same degree as the optimal demodulators when the number of words (n) is increased. Consequently, two of the suboptimal demodulators investigated were able to use a large number of words easier than the optimal demodulators could use two words, and therefore get better performance than the optimal demodulators with less complexity.

All of the results of the first five chapters are restricted by the three main assumptions made in the introduction, namely, the assumption of Gaussian, bandlimited, white noise, the assumption of independent additive noise, and the assumption of Gaussian data. Without the first two assumptions the derivation of the expression for $f_{Y_j|z}(k\,|\,z(t))$ becomes far more complex and may not even be possible. Hence the analytic description of the optimal demodulators may be very complex or impossible. This does not necessarily mean that a high statistical dependence between data samples could not be used to improve the demodulation process, but schemes of using this dependence will not be as obvious. However, if some method of doing this is invented, it may be possible to evaluate the performance of the demodulator by an application of a Monte Carlo simulation on a digital computer similar to that used in the restrictive case considered in the first five chapters. All that is necessary is a method of generating signal and noise samples from the proper distributions and a program that simulates the operation of the demodulator in discrete time.

As was pointed out in the introduction, the assumption of band-limited Gaussian noise is a fairly good one. Maintaining this assumption then, the case of multiplicative noise (random bias in the transmitter or the channel) could be analyzed by Monte Carlo methods simply by forming the proper combination of signal and noise in the computer before the simulated demodulation. Again, the methods of using high correlation between data samples in the demodulation process may not be apparent, but a Monte Carlo simulation may give some insight into this. Simple suboptimal schemes such as Rauch's demodulation method may show an improved performance over present day methods.

The assumption of Gaussian data was not used in the derivation of the expressions for the optimal or suboptimal demodulators, but rather the data distribution entered into the demodulator description as the weighting by $f_Y(Y_1, \ldots, Y_{j-1}, k, Y_{j+1}, \ldots, Y_n)$ or $f_Y(Y_1, \ldots, Y_{j-1}, k)$ or, in the case of Rauch's suboptimal demodulator, not at all. Certainly the specific results of Chapters 4 and 5 are good only for Gaussian data, but the demodulation schemes (both optimal and suboptimal) should be equally valid for data from other distributions, and if expressions for the multivariant probability density function of the data are available, the specific performance of the demodulators could be obtained by the same Monte Carlo simulation (or direct computation in the case of the present day demodulator and Rauch's suboptimal demodulator) as used in Chapters 1 through 5.

In addition to extending the analysis of optimal and suboptimal binary PCM demodulators to other forms of noise and to data from other than a Gaussian distribution, several other extensions appear worthwhile.

Although Smith's suboptimal demodulator did not give an improvement in performance worthy of the added complexity over the present day, bit-by-bit correlation demodulators, this does not mean that the idea should be abandoned. Some weighting on the influence of $\hat{Y}_1$ may give better results for the case of very high correlation between data samples. This, however, would result in some added complexity and the robustness characteristics of the demodulator for changes in ρ might not be desirable. Combining Smith's demodulation scheme with Rauch's scheme is another possibility for improved demodulation. This is equivalent to substituting $f_{Y_j|z_jY}(k \mid z_j, \hat{Y}_1, \ldots, \hat{Y}_{j-1})$ for $f_Y(k)$ in suboptimal number three. The added complexity over suboptimal number 3 is slight, and for the higher values of S/N (where $\hat{Y}_1, \ldots, \hat{Y}_{j-1}$ are very accurate) the improvement in the demodulation may be worthwhile. This could be evaluated with a Monte Carlo simulation as was done with suboptimal number three.

Improved computation schemes or faster computers would allow a more accurate determination of the performance (and the optimal weightings for the last three bits) of Rauch's suboptimal demodulator for six-bit data. However, the weightings on the first three bits (which are the same as in the three-bit case) have the most pronounced effect on the MAE and RMSE, so that approximate results of Chapter 5 should give a very good indication of the demodulator performance. The improvement that could be gained by considering more than 5 words in the demodulation with Rauch's demodulator would be a meaningful extension, but it is doubtful that the direct computation would be feasible due to the difficulty in evaluating $f_Y(Y_1, Y_2, Y_3, Y_4, Y_5)$. However, the amount of improvement could be evaluated approximately with a Monte Carlo simulation.

For Gaussian data the probability that $g_1(t)$ was transmitted in any bit position is .5, so weighting by this probability in Rauch's demodulator offers no benefit. However, if the data distribution is skewed, then weighting G_{1r} and G_{2r} in equation 5-7 by the probability that $y_{jr} = g_1(t)$ and $y_{jr} = g_2(t)$ should improve the demodulation, particularly for r = 1 or 2 where the improvement has the most effect on the MAE and RMSE.

The extensions mentioned above may improve the suboptimal demodulators discussed in Chapter 5, but the amount of improvement would not be expected to be substantial. Some entirely new suboptimal demodulation scheme may be invented that would give good performance and yet be practical to build, and certainly this possibility should not be dismissed. However, Rauch's suboptimal demodulator seems to be the only logical way to take advantage of a high statistical dependence between nearby data samples when the demodulation is accomplished bit by bit. And suboptimal number 3 is the logical extension of Rauch's scheme to word-by-word demodulation.

Up to this point we have not discussed the possibility of the use of a high correlation between nearby PCM words after one-word bit-by-bit demodulation. This would allow demodulation with equipment now on hand and improvement at some later time. This post-demodulation improvement of the demodulated data could be done in two ways, bit by bit or word by word. In bit-by-bit demodulation improvement, the demodulated bits in a particular bit position would be compared over successive words to improve the demodulation of that bit. However, it is seen that this is almost equivalent to Rauch's demodulation scheme and in fact, can do no better than Rauch's demodulator, and therefore offers nothing very new except the ability to improve the demodulation while still using present day equipment. In word-by-word post-demodulation improvement, the demodulated words would be compared in some manner (and possibly weighted in some way by $f_Y(Y_1, \ldots, Y_n)$) to improve the demodulation of Y_j. A simple yet possibly efficient way of doing this would be to use a weighted average (as in Rauch's demodulator) of the successive demodulated words to determine the new demodulated word. The extension of this to a somewhat more complex but possibly more efficient method would be to then weight (as in suboptimal number 3) this weighted average by $f_Y(Y_j)$. The evaluation of these schemes of post-demodulation improvement might possibly be done directly, but in any event could certainly be done by Monte Carlo methods.

It appears that unless a radically different demodulation scheme is hit upon the most fruitful extension of this dissertation is in the area of post-demodulation improvement.

Appendix I

CORRELATION COEFFICIENTS FOR BUTTERWORTH DATA

mth order Butterworth data is data that results from the passage of white noise through a Butterworth filter and has an essentially flat power spectrum out of some break frequency, f_I, and beyond f_I the roll-off rate is 6 db per octave for first order data, 12 db per octave for second order data, etc. For mth order Butterworth data, the spectrum is [M3]

$$S(f) = \frac{1}{1 + \left(\frac{f}{f_I}\right)^{2m}}$$

Letting $x = 2\pi \frac{f_I}{f_s}$ where f_s is the sampling frequency, McRae and Smith [M4] derived the following expressions for $\rho(\tau)$, where $\tau = 1/f_s$.

For first order data

$$\rho(\tau) = \exp(-x)$$

For second order data

$$\rho(\tau) = \sqrt{2}\ \exp(-x/\sqrt{2})\ \cos(x/\sqrt{2} - \pi/4)$$

For third order data

$$\rho(\tau) = (1/2)\ \exp(-x) + \exp(-x/2)\ \cos(.866x - \pi/3)$$

McRae has tabulated the required sampling rates for various percentage errors [M3]. If we specify a sampling rate such that the interpolation error does not exceed the quantization error from the coding of the analog data into PCM, then the required sampling rates are those from McRae's tables for the quantization error.

If we assume that the quantization error is uniformly distributed from -1/2 to 1/2, i.e.,

$$f(x) = \begin{cases} 1, & -1/2 \leqq x \leqq 1/2 \\ 0, & \text{otherwise} \end{cases}$$

then

$$\text{MAE} = \int_{-\infty}^{\infty} |x| f(x)\, dx = 2\int_{0}^{1/2} x dx = \frac{1}{4}$$

$$\text{MSE} = \int_{-\infty}^{\infty} x^2 f(x)\, dx = \int_{-1/2}^{1/2} x^2 dx = \frac{1}{12}$$

$$\text{RMSE} = \sqrt{1/12} = .2887$$

If we normalize the RMSE by the peak-to-peak signal, then

$$\frac{\text{RMSE}}{\text{P - P Signal}} = .04124 \text{ for 3-bit words}$$

$$\frac{\text{RMSE}}{\text{P - P Signal}} = .00458 \text{ for 6-bit words}$$

In reference M3, McRae uses the RMSE over the RMS signal to get the percent error. Assuming sinusoidal data, the RMS signal equals the peak-to-peak signal divided by $2\sqrt{2}$. Multiplying the above values by $2\sqrt{2}$ we have the following normalized errors:

.1167 or about 10% for 3-bit words

.01296 or about 1% for 6-bit words.

Using these values, the sampling rates from reference M3, and the expressions for ρ, table A-1-1 was prepared. It gives the correlation coefficient $\rho_{12}(\tau)$, between adjacent PCM words and the correlation coefficient, $\rho_{13}(\tau)$, between PCM words i and i + 2.

TABLE A-1-1

Correlation Coefficients for Butterworth Data

1% Data (6 bit)

Order of Data	Interpolation Method	f_s/f_I	$\rho_{12}(\tau)$	$\rho_{13}(\tau)$
1	Ideal	11500	.99945	.99891
2	Linear	40	.98855	.95756
3	Linear	19	.97311	.89714

10% Data (3 bit)

Order of Data	Interpolation Method	f_s/f_I	$\rho_{12}(\tau)$	$\rho_{13}(\tau)$
1	Ideal	125	.95098	.90436
2	Linear	8.1	.79433	.44942
3	Linear	5.8	.74757	.28593

Appendix II

GENERALIZATION TO ARBITRARY BIT WAVEFORMS

Smith [S2] generalizes from equal energy signals with a correlation coefficient of -1 to two arbitrary signals $f_1(t)$ and $f_2(t)$ as follows. A "correlation parameter", α, is defined by

$$\alpha = \frac{2\int_{T_B} f_1(t)\, f_2(t)\, dt}{\int_{T_B} (f_1^2(t) + f_2^2(t))dt}$$

Let

$$g_1(t) = \frac{1}{2}(f_1(t) - f_2(t))$$

$$g_2(t) = -\frac{1}{2}(f_1(t) - f_2(t)) \tag{A-2-1}$$

$g_1(t)$ and $g_2(t)$ have equal energies and a λ of

$$\lambda = \frac{1}{T_B S^2}\int_{T_B} g_1(t)\, g_2(t)\, dt = \frac{-1}{4T_B S^2}\int_{T_B} (f_1(t) - f_2(t))^2 dt$$

Since $S^2 = \frac{1}{T_B}\int_{T_B} g_1(t)^2 dt$

$$= \frac{1}{4T_B}\int_{T_B} (f_1(t) - f_2(t))^2 dt$$

we have $\lambda = -1$.

If the transmitted waveforms are $f_1(t)$ and $f_2(t)$, then

$$z_{ir} = f_{ir}(t) + n_{ir}(t)$$

If $\frac{1}{2}(f_1(t) + f_2(t))$ is subtracted from $z_{ir}(t)$, then

$$z'_{ir}(t) = f_{ir}(t) + n_{ir}(t) - \frac{1}{2}(f_1(t) + f_2(t))$$

$$= g_{ir}(t) + n_{ir}(t)$$

and the results of the computer simulation for signals of $\pm$ S/N apply. Since this is a reversible linear transformation, the results based on the S/N for the g(t) waveforms are the same results that we could get for the original S/N for the f(t) waveforms. The results will still be optimal for the optimal demodulators due to the linearity of the transformation. The conversion factor is then the square root of the ratio of the signal power, S_f^2, in the f(t) waveforms to the signal power, S_g^2, in the g(t) waveforms. This becomes

$$\frac{S_f}{S_g} = \left[\frac{\frac{1}{2}\frac{1}{T_B}\int_{T_B}(f_1^2(t) + f_2^2(t))dt}{\frac{1}{4T_B}\int_{T_B}(f_1(t) - f_2(t))^2\,dt}\right]^{1/2}$$

$$= \sqrt{\frac{2}{1-\alpha}}$$

or

$$(S/N)_f = \sqrt{\frac{2}{1-\alpha}}\;(S/N)_g \qquad \text{(A-2-2)}$$

The use of equation A-2-2 allows the application of the results of Chapters 4 and 5 to any arbitrary bit waveforms used in the binary PCM transmission.

Appendix III

COMPUTER PROGRAMS

The computer programs were written in FORTRAN II [M2] for the University of Michigan Executive System, which utilizes an IBM 7090 computer. Because the Michigan Executive System utilizes other compilers which use the same library functions as the FORTRAN compiler, some of these functions are written without a terminal "F" in FORTRAN. Examples of this are the "SIN", "EXP", and "SQRT" functions. Also the input/output statements used in the programs are necessitated by the Michigan Executive System's use of off-line cardreaders and printers.

The basic computer programs used are presented in Tables A-3-1 through A-3-5. The comment statements (a "C" in column 1) give an explanation of what each part of the program does, which includes the meaning of the important variable names that are used.

Table A-3-1 shows the program used to compute the MAE and RMSE for the present day, bit-by-bit, one-word demodulator. The bit-error probability for each S/N was computed by reading the probability that a normal (0, 1) random variable is greater than S/N from standard probability tables. This is then read in as "PR" in the program. This program, as were most of the programs, was written so that any size words could be used, but for m = 3 the values of $f_Y(k)$ (P(I) in the program) were read in (from probability tables), instead of being computed in the program, for more accuracy.

Table A-3-2 shows the general program for the three optimal demodulators, and is generalized for any size words and up to 3 words (n = 3) used in the demodulation. The results of all three optimal demodulators are computed at the same time, since equations 1-1, 1-2 and 1-3 all use the values of $f_{Y_j|Z}(k|z(t))$ (P(I) in the program). The subroutine for generating Gaussian random numbers was a Michigan Executive System library function, and was called as RANDND (σ_Y, m_Y, RNO), where RNO is the starting number for generating the sequence of random numbers. For a random start, RNO is set to 0, as was done in all the programs which used random numbers. The programs that were used to compute the bias and robustness of the optimal demodulators are not shown since they are obvious modifications of the program in Table A-3-2.

Table A-3-3 shows the program which was used to estimate the performance of Smith's suboptimal demodulator. As was mentioned in Chapter 5 four demodulations or subiterations were made for each iteration to insure that the starting effect of not having $\hat{Y}_1$ available was smoothed out.

Table A-3-4 gives the program which computes the MAE and RMSE for Rauch's suboptimal demodulator. As was discussed in Chapter 5, a Monte Carlo simulation was not used. Rather, these performance parameters were computed directly. To compute the probability of a certain sequence of transmitted bits occurring in the rth bit of the sequence of transmitted PCM words, denoted by PBS_{ir}, it was first necessary to compute the probability of a certain sequence of PCM words occuring, $f_Y(Y_1, Y_2, Y_3, Y_4, Y_5)$, and then add together all the f_Y's that would give a certain PBS_{ir}. As pointed out in Chapter 5,

$$f_Y(Y_1, Y_2, Y_3, Y_4, Y_5) = \frac{f_Y(Y_1, Y_2, Y_3) f_Y(Y_3, Y_4, Y_5)}{f_Y(Y_3)}$$

To get the necessary accuracy in the computation of the PBS_{ir}'s, it was necessary to integrate $f_X(x_1, x_2, x_3)$ over the region where each x_i quantizes to the particular value of each Y_i. Integration[1] by Simpson's 1/3 Rule [M1] was used, with the intervals of integration successively

[1] The program that does the integration (subprograms CALC1, CALC2, and CALC3 in Table A-3-4) was developed by Capt. Edward G. Preston.

split until the desired accuracy was reached (EP1 in the program) between successive evaluations. The values of $f_Y(Y_3)$ were read in from standard probability tables. The optimal weightings, w_{ir}, were determined by computing the bit-error probability (PBE_r) for each possible set of w_{ir}'s (no finer breakdown of the w_{ir}'s than .01 was made) and then selecting the set that gave the smallest PBE_r. Using the PBE_r's, the probability of an error of i was computed, and, as in the program of Table A-3-1, this was used to compute the MAE and RMSE.

Table A-3-5 gives the program which uses a Monte Carlo simulation to estimate the performance of suboptimal demodulator number 3. The weightings, w_{ir}, from the previous program's results, were read in and used in this program.

These 5 programs are presented here for the benefit of anyone who desires to extend the work done in this dissertation. The author does not claim to be an expert programmer, and there are undoubtedly many small improvements that could be made in these programs to improve their accuracy and efficiency. It should be noted that some efficiency is lost in the generalization of the programs to a general m (and n in the optimal demodulator program), but this is more than made up for the programmer in the time spent in the program preparation.

Table A-3-1. Computer Program, Present Day Demodulator

```
      DIMENSION JC(64,6),P(64),PP(64,6),PW(6),PE(64)
C     BYPASSES UNDERFLOW ERROR.
      CALL FTRAP
C     SETS CONSTANTS. MM=m, SY=σY, XMY=mY, MM1=2^m,
C
      READ INPUT TAPE 7,101,MM,SY,XMY
      MM1=2**MM
      MM2=MM1-1
C     COMPUTES P(I)=fY (I).
      PS=0.
      DO 2 I=1,MM1
      X=I-1
      P(I)=EXP (-.5*((X-XMY)/SY)**2)
 2    PS=PS+P(I)
      DO 3 I=1,MM1
 3    P(I)=P(I)/PS
C     COMPUTES JC(I,K)=yIK ÷ S/N.
      DO 130 K=1,MM
 130  JC(1,K)=-1
      DO 162I=1,MM2
      DO 140 K=1,MM
 140  JC(I+1,K)=JC(I,K)
      DO 150 K=1,MM
      N=MM-K+1
      JC(I+1,N)=JC(I+1,N)+1
      IF(JC(I+1,N)) 151,151,150
 151  JC(I+1,N)=1
      GO TO 162
 150  JC(I+1,N)=-1
 162  CONTINUE
C     COMPUTES PP(J,K)=SUM OF THE PROBABILITIES THAT THE Y'S WERE
C     TRANSMITTED SUCH THAT K BITS IN ERROR WOULD GIVE AN ABSOLUTE
C     ERROR OF J.
      DO 4 I=1,MM1
      DO 4 K=1,MM
 4    PP(I,K)=0.
      DO 20 J=1,MM2
      M=MM1-J
      DO 20 I=1,M
      N=I+J
      JJ=0
      DO 10 K=1,MM
 10   JJ=JJ+(XABSF(JC(I,K)-JC(N,K)))/2
      PP(J,JJ)=PP(J,JJ)+P(I)*P(N)
 20   CONTINUE
C     SN=S/N,PR=BIT-ERROR PROBABILITY,XMAE=MAE,XMSE=MSE,
 70   READ INPUT TAPE 7,1,PR,SN
      XMAE=0.
      XMSE=0.
```

Table A-3-1 (Continued)

```
C  COMPUTES PE(I)=THE PROBABILITY OF AN ABSOLUTE ERROR OF I.
C  PW(I)=THE PROBABILITY THAT I BITS ARE IN ERROR.
      DO 5 I=1,MM
 5    PW(I)=PR**I*(1.-PR)**(MM-I)
      DO 6 I=1,MM1
 6    PE(I)=0.
      DO 30 I=1,MM2
      DO 30 K=1,MM
 30   PE(I)=PE(I)+PP(I,K)*PW(K)
C  COMPUTES AND PRINTS XMAE=MAE AND RMSE=RMSE.
      DO 40 I=1,MM2
      XI=I-1
      XMAE=XMAE+XI*PE(I)
 40   XMSE=XMSE+XI**2*PE(I)
      RMSE=SQRT(XMSE)
      WRITE OUTPUT TAPE 6,50,SN,PR,XMAE,RMSE
C  RETURNS TO READ ANOTHER S/N AND PR.
      GO TO 70
 1    FORMAT(10F10.7)
 50   FORMAT(2X,4HS/N=F6.4,2X,13HP(BIT ERROR)=F12.7,2X,4HMAE=F12.7,
     1  2X,5HRMSE=F12.7)
 51   FORMAT(1H1)
 100  FORMAT(6(2X,I4))
 101  FORMAT(I2,2F8.4)
      END
```

Table A-3-2. Computer Program, Optimal Demodulators

```
      DIMENSION JC(64,6),ER(3),ERSR(3),ERS(3),ERSS(3),XM(64),XMS(64),
     1WT(64,64),W(64),WW(64),SYM(3),X(3),JYT(3),CU(3,6),PWR(64,3),P(64),
     1YH(3),HAA(3),HBB(3),ERR(3),SERR(3),SD1(3),SD2(3),Q(3),PC(3),PCC(3)
C  BYPASSES UNDERFLOW ERROR.
      CALL FTRAP
C  SETS CONSTANTS. MM=m,NN=n,MM1=2^m,XMY=m_Y,SY=σ_Y.
      RNO=0.
      RN2=0.
 69   READ INPUT TAPE 7,11,MM,AA,NN
      XMM=MM
      XNN=NN
      MM1=2**MM
      MM2=MM1-1
      XMM2=MM2
      XMY=XMM2/2.
      SY=XMY/AA
      RSY=1./SY
C  COMPUTES JC(I,K)=y_IK ÷ S/N.
      DO 30 K=1,MM
 30   JC(1,K)=-1
      DO 62 I=1,MM2
      DO 40 K=1,MM
 40   JC(I+1,K)=JC(I,K)
      DO 50 K=1,MM
      N=MM-K+1
      JC(I+1,N)=JC(I+1,N)+1
      IF(JC(I+1,N)) 51,51,50
 51   JC(I+1,N)=1
      GO TO 62
 50   JC(I+1,N)=-1
 62   CONTINUE
C  SETS CONSTANTS. LL=NO.OF ITERATIONS,SN=S/N,RH=ρ,
C  A=σ AND B=m FOR NOISE SAMPLES.
 70   READ INPUT TAPE 7,13,LL,SN,RH
      IF(LL) 69,69,68
 68   CONTINUE
      READ INPUT TAPE 7,14,A,B
      A2=1.-A**2
      A3=1./(2.*A**2)
      G1=A**(NN*MM)*EXP (A3*XMM*XNN*B**2)
      RRSC=1.0/(1.0-RH**2)
      RRSCR=SQRT (RH*RRSC)
      RRSC2=0.5*RRSC
C  COMPUTES WT(I,M)=f_Y(I,M) ,WW(I)=f_Y(I) ,W(I)=1/f_Y(I),
C
      GO TO (74,73,73),NN
 73   DO 72 I=1,MM1
      Y=I-1
      X=(Y-XMY)*RSY
```

Table A-3-2 (Continued)

```
      XM(I)=X*RRSCR
   72 XMS(I)=X**2*RRSC2
      DO 75 I=1,MM1
      DO 75 M=1,MM1
 75   WT(I,M)=EXP (XM(I)*XM(M)-XMS(I)-XMS(M))
  74  DO 76 I=1,MM1
      Y=I-1
      W(I)=EXP (.5*((Y-XMY)*RSY)**2)
 76   WW(I)=1./W(I)
C  XM=m AND SYM=σ FOR x DISTRIBUTION.
      XM(1)=XMY
      SYM(1)=SY
      SYM(2)=SY/SQRT (RRSC)
      SYM(3)=SYM(2)
      A9=XMY*(1.-RH)
      DO 71 I=1,3
      ER(I)=0.
      ERSR(I)=0.
      ERS(I)=0.
 71   ERSS(I)=0.
C  INITIATES LL ITERATIONS
      DO 500 L1=1,LL
C  FORMS  Y_1,...,Y_n. X(1)=x_2, X(2)=x_1, X(3)=x_3, JYT(1)=Y_2, JYT(2)=Y_1,
C
C  JYT(3)=Y_3.
C
      USS=0.
      USSS=0.
      DO 140 I=1,NN
      X(I)=RANDND(0.,1.,RNO)*SYM(I)+XM(I)
      XM(I+1)=A9+RH*X(1)
      JYT(I)=X(I)+.5
      IF(JYT(I)) 99,130,100
   99 JYT(I)=0
      GO TO 130
 100  IF(JYT(I)-MM2) 130,130,110
 110  JYT(I)=MM2
  130 J=JYT(I)+1
C  FORMS z_ir  AND WEIGHTING. US=ν_IK, CU(I,K)=z_IK , G=WEIGHTING
C
C  DUE TO MODIFIED NOISE DISTRIBUTION.
      DO 140 K=1,MM
      US=RANDND(0.,1.,RN2)*A+B
      IF(JC(J,K)) 136,136,138
  136 CU(I,K)=US-SN
      GO TO 139
 138  CU(I,K)=SN-US
 139  USS=USS+US
 140  USSS=USSS+US**2
      G=G1*EXP ((A2*USSS-2.*B*USS)*A3)
C
C  COMPUTES f_{Y_j|Z}(k|z(t))      AND Ŷ_1 , PWR(I,L) = Σ_{r=1}^{m} z_Ir y_I(L)r ÷ S/N.
C
```

Table A-2-3 (Continued)

```
C   PPR1=70.-MAXIMUM PWR(I,1)·S/N AND IS USED TO AVOID VERY
C   SMALL NUMBERS IN THE COMPUTER AND IMPROVE ACCURACY,
C   P(I)=f_{Y_j|z}(I|z(t))      , YH1=Ŷ_1 ,
C
      PPR1=0.0
      DO 170 I=1,MM1
      DO 144 L=1,NN
 144  PWR(I,L)=0.
      DO 160 K=1,MM
      IF(JC(I,K)) 145,145,150
 145  DO 146 L=1,NN
 146  PWR(I,L)=PWR(I,L)-CU(L,K)
      GO TO 160
 150  DO 151 L=1,NN
 151  PWR(I,L)=PWR(I,L)+CU(L,K)
 160  CONTINUE
      IF(PPR1-PWR(I,1)) 170,170,165
 165  PPR1=PWR(I,1)
 170  CONTINUE
      PPR1=70.-PPR1*SN
      DO 171 I=1,MM1
 171  P(I)=WW(I)*EXP (PWR(I,1)*SN+PPR1)
      GO TO (180,172,172),NN
 172  DO 174 I=1,MM1
      PP=0.
      DO 173 J=1,MM1
 173  PP=PP+WT(J,I)*EXP (PWR(J,I)*SN)
 174  P(I)=P(I)*W(I)*PP
      GO TO (180,180,175),NN
 175  DO 177 I=1,MM1
      PP=0.
      DO 176 J=1,MM1
 176  PP=PP+WT(I,J)*EXP (PWR(J,3)*SN)
 177  P(I)=P(I)*W(I)*PP
 180  PP=0.
      PS=0.
      DO 181 I=1,MM1
      PS=PS+P(I)
      IF(P(I)-PP) 181,181,182
 182  PP=P(I)
      YH(1)=I-1
 181  CONTINUE
      DO 499 I=1,MM1
 499  P(I)=P(I)/PS
C   COMPUTES YH(2)=Ŷ_2 .
      PPP=0.
      DO 190 I=1,MM1
      PPP=PPP+P(I)
      IF(PPP-.5) 190,191,191
 190  CONTINUE
 191  YH(2)=I-1
```

Table A-3-2 (Continued)

```
C   COMPUTES YH(3)=\hat{Y}_3 .
        YH(3)=0.
        DO 192 I=1,MM1
        D2=I-1
 192    YH(3)=YH(3)+D2*P(I)
C   COMPUTES AE_2 OR AE_4 AND SE_2 OR SE_4 AND CUMULATIVE SUMS.
C
C   ER(I)= \sum_{k=1}^{L} (AE_j)_k   ,ERS(I)= \sum_{k=1}^{L} (SE_j)_k
C
C   ERSR(I)= \sum_{k=1}^{L} (AE_j)_k^2   ,ERSS(I)=\sum_{k=1}^{L} (SE_j)_k^2   , I REFERS TO THE
C
C   PARTICULAR DEMODULATOR.
        DO 502 I=1,3
        HAA(I)=0.
        HBB(I)=0.
        DO 501 K=1,MM1
        XK=K-1
        XX1=ABSF(XK-YH(I))
        HAA(I)=HAA(I)+XX1*P(K)
 501    HBB(I)=HBB(I)+(XX1**2)*P(K)
        HAA(I)=HAA(I)*G
        HBB(I)=HBB(I)*G
        ER(I)=ER(I)+HAA(I)
        ERS(I)=ERS(I)+HBB(I)
        ERSR(I)=ERSR(I)+HAA(I)**2
 502    ERSS(I)=ERSS(I)+HBB(I)**2
C   STOPS ITERATIONS AND PRINTS HEADINGS.
 500    CONTINUE
        WRITE OUTPUT TAPE 6,5
        XLL=LL
        WRITE OUTPUT TAPE 6,20,NN
        WRITE OUTPUT TAPE6,6,MM
        WRITE OUTPUT TAPE6,3
        WRITE OUTPUT TAPE6,2,SN,RH,LL
        WRITE OUTPUT TAPE 6,12,B,A
        WRITE OUTPUT TAPE6,3
C   COMPUTES AND PRINTS RESULTS. ERR(I)=M\hat{A}E_I  ,SD1(I)=2\hat{\sigma}_{M\hat{A}E_I}
C
C   SERR(I)=M\hat{S}E_I  ,SD2(I)=2\hat{\sigma}_{M\hat{S}E_I}  ,Q(I)=R\hat{M}SE_I  ,PC(I)=% FOR 2\hat{\sigma}_{M\hat{A}E_I}
C
C   PCC(I)= % FOR 2\hat{\sigma}_{M\hat{S}E_I}
C
        DO 201 I=1,3
        ERR(I)=ER(I)/XLL
        SERR(I)=ERS(I)/XLL
        SD1(I)=2.*SQRT ((ERSR(I)/XLL-ERR(I)**2)/XLL)
        SD2(I)=2.*SQRT ((ERSS(I)/XLL-SERR(I)**2)/XLL)
        Q(I)=SQRT (SERR(I))
        PC(I)=SD1(I)/ERR(I)*100.
        PCC(I)=SD2(I)/SERR(I)*100.
```

Table A-3-2 (Continued

```
      GO TO(203,204,205),I
 203  WRITE OUTPUT TAPE 6,15
      GO TO 206
 204  WRITE OUTPUT TAPE 6,17
      GO TO 206
 205  WRITE OUTPUT TAPE 6,18
 206  WRITE OUTPUT TAPE 6,16,ERR(I),SD1(I),SERR(I),SD2(I),Q(I)
      ERR(I)=ERR(I)+SD1(I)
      Q(I)=SQRT (SERR(I)+SD2(I))
      WRITE OUTPUT TAPE 6,21,ERR(I),Q(I)
      ERR(I)=ERR(I)-SD1(I)*2.
      SERR(I)=SERR(I)-SD2(I)
      IF(SERR(I)) 207,207,208
 207  Q(I)=0.
      GO TO 209
 208  Q(I)=SQRT (SERR(I))
 209  WRITE OUTPUT TAPE 6,21,ERR(I),Q(I)
 201  WRITE OUTPUT TAPE 6,19,PC(I),PCC(I)
C  RETURNS TO READ ANOTHER SET OF DATA.
      GO TO 70
 2    FORMAT(3HSN=F5.3,2X,4HRHO=F5.3,2X,15HNO, ITERATIONS=I4)
    3 FORMAT (1H0)
    5 FORMAT (1H1)
 6    FORMAT(2X,I2,1X,9HBIT WORDS)
 11   FORMAT(I2,F12.7,I1)
 12   FORMAT(35HMODIFIED NOISE DISTRIBUTION   MEAN=F8.4,2X,3HSD=F8.4)
 13   FORMAT (I4,2F8.4)
 14   FORMAT(8F10.6)
 15   FORMAT(23HMINIMUM PE  DEMODULATOR)
 16   FORMAT(4HMAE=F12.7,2X,4H2SD=F12.7,2X,4HMSE=F12.7,2X,4H2SD=F12.7,
     1  5X,5HRMSE=F12.7)
 17   FORMAT(23HMINIMUM MAE DEMODULATOR)
 18   FORMAT(23HMINIMUM MSE DEMODULATOR)
 19   FORMAT(8HPERCENT=F8.4,25X,8HPERCENT=F8.4)
 20   FORMAT(22HOPTIMAL DEMODULATION- ,I1,6H WORDS)
 21   FORMAT(4X,F12.7,63X,F12.7)
      END
```

Table A-3-3. Computer Program, Smith's Demodulator

```
      DIMENSION SU(6),ASE(4),Q(4),JC(64,6),VS(5,6),SDMSE(4),P(64)
      DIMENSION AES(4),SDMAE(4),PCC(4),PC(4),XS2(64),SES(4),SESS(4)
      DIMENSION Z(5)
C  BYPASSES UNDERFLOW ERROR.
      CALL FTRAP
C  SETS CONSTANTS. MM=m,MM1=2^m, XMY=m_Y,SY=σ_Y.
      RNO=0.
      RN1=0.
 400  READ INPUT TAPE 7,11,MM,AA
      MM1=2**MM
      XMM1=MM1
      XMM=MM
      MM2=MM1-1
      MM3=MM+1
      XMM2=MM2
      XMY=XMM2/2.
      SY=XMY/AA
      RSY=1./SY
C  COMPUTES JC(I,K)=y_IK ÷ S/N.
      DO 30 K=1,MM
 30   JC(1,K)=-1
      DO 62 I=1,MM2
      DO 40 K=1,MM
 40   JC(I+1,K)=JC(I,K)
      DO 50 K=1,MM
      N=MM-K+1
      JC(I+1,N)=JC(I+1,N)+1
      IF(JC(I+1,N)) 51,51,50
 51   JC(I+1,N)=1
      GO TO 62
 50   JC(I+1,N)=-1
 62   CONTINUE
C  COMPUTES XS2(I)=1/f_Y(I).
      DO 63 I=1,MM1
      Y=I-1
 63   XS2(I)=.5*(((Y-XMY)*RSY)**2)
C  SETS CONSTANTS. LL=NO. OF ITERATIONS, SN=S/N,RH=ρ12 ,RHH=ρ13
 70   READ INPUT TAPE 7,13,LL,SN,RH,RHH
      IF (LL) 400,400,701
 701  JCT=0
      F=RH**2
      E=1.-F
      SYM=SY*SQRT (E)
      E=1./E
      D=(1.-RH)*XMY
      T=RSY**2*.5*E
      DO 71 I=1,4
      AES(I)=0.
```

Table A-3-3 (Continued)

```
      SES(I)=0.
 71   SESS(I)=0.
      XLL=LL
C  INITIATES LL ITERATIONS
      DO 500 L=1,LL
C  GENERATES THE NEEDED GAUSSIAN RANDOM NUMBERS
      DO 600 I=1,4
      Z(I)=RANDND(0.,1.,RN0)
      DO 600 J=1,MM
 600  VS(I,J)=RANDND(0.,1.,RN1)
C  SYM1=σ AND XM1=m FOR x DISTRIBUTION.
      SYM1=SY
      XM1=XMY
C  INITIATES 4 SUBITERATIONS.
      DO 500 M=1,4
C  COMPUTES X= x2 AND Y2=Y2.
      X=SYM1*Z(M)+XM1
      J=X+1.5
      IF(J-1) 100,131,111
 100  J=1
      GO TO 131
 111  IF(J-MM1) 131,131,120
 120  J=MM1
 131  Y2=J-1
C  COMPUTES SU(K)= z2K.
      DO 150 K=1,MM
      US=VS(M,K)
      IF(JC(J,K)) 144,144,146
 144  SU(K)=US-SN
      GO TO 150
 146  SU(K)=US+SN
 150  CONTINUE
C  COMPUTES P(I)= fY2|z2Y1(I|z2,Ŷ1).
      PS=0.
      DO 170 I=1,MM1
      U=0.
      DO 160 K=1,MM
      IF(JC(I,K)) 154,154,156
 154  U=U-SU(K)
      GO TO 160
 156  U=U+SU(K)
 160  CONTINUE
      GO TO (162,163,163,163,163),M
 162  V=XS2(I)-U*SN
      GO TO 169
 163  XI=I-1
      V=(XI-XNH)**2*T-U*SN
 169  P(I)=EXP (-V)
 170  PS=PS+P(I)
      DO 180 I=1,MM1
```

Table A-3-3 (Continued)

```
 180   P(I)=P(I)/PS
C  COMPUTES YH=$\hat{Y}_2$
       YH=0.
       DO 190 I=1,MM1
       D2=I-1
 190   YH=YH+D2*P(I)
C
C  COMPUTES AND STORES AES(M)=$\sum_{k=1}^{L}(AE_1)_k$ ,SES(M)=$\sum_{k=1}^{L}(AE_1)_k^2$,
C
C  SESS(M)= $\sum_{k=1}^{L}(AE_1)_k^4$   , M=SUBITERATION.
C
C  SETS $\sigma_x$ AND $m_x$ FOR THE NEXT SUBITERATION.
       H=ABSF(Y2-YH)
       AES(M)=AES(M)+H
       H=H**2
       SES(M)=SES(M)+H
       SESS(M)=SESS(M)+H**2
       XNH=D+RH*YH
       SYM1=SYM
 500   XM1=D+RH*X
C  COMPUTES AND PRINTS AES(I)=$\hat{MAE}_I$ ,ASE(I)=$\hat{MSE}_I$,SDMAE(I)=2$\hat{\sigma}_{\hat{MAE}_I}$
C
C  SDMSE(I)=2$\hat{\sigma}_{\hat{MSE}_I}$,PCC(I)= % FOR 2$\hat{\sigma}_{\hat{MAE}_I}$,PC(I)= % FOR 2$\hat{\sigma}_{\hat{MSE}_I}$
C
C  Q(I)=RM$\hat{SE}_I$  I REFERS TO THE SUBITERATION.
       DO 201 I=1,4
       AES(I)=AES(I)/XLL
       ASE(I)=SES(I)/XLL
       SDMAE(I)=2.*SQRT ((ASE(I)-AES(I)**2)/XLL)
       PCC(I)=SDMAE(I)/AES(I)*100.
       SDMSE(I)=2.*SQRT ((SESS(I)/XLL-ASE(I)**2)/XLL)
       Q(I)=SQRT (ASE(I))
 201   PC(I)=SDMSE(I)/ASE(I)*100.
       WRITE OUTPUT TAPE 6,5
       WRITE OUTPUT TAPE 6,6,MM
       WRITE OUTPUT TAPE 6,3
       WRITE OUTPUT TAPE 6,7
       WRITE OUTPUT TAPE6,17,SN,RH,LL
       WRITE OUTPUT TAPE 6,14
       DO 202 I=1,4
 202   WRITE OUTPUT TAPE 6,15,I,AES(I),SDMAE(I),PCC(I),ASE(I),
      1 SDMSE(I),PC(I),Q(I)
C  RETURNS TO READ ANOTHER SET OF DATA.
       GO TO 70
 1     FORMAT(6I6)
 2     FORMAT(3HSN=F5.3,2X,4HRHO=F5.3,2X,6HRHO13=F5.3,,2X,
      1    15HNO. ITERATIONS=I4)
 3     FORMAT(1H0)
 4     FORMAT(8E14.6)
 5     FORMAT(1H1)
```

Table A-3-3 (Continued)

```
6     FORMAT(2X,I2,1X,9HBIT WORDS)
7     FORMAT(28HSUBOPTIMAL DEMODULATOR NO, 2)
8     FORMAT(2X,I2,4X,4HMSE=F12.7,2X,4H2SD=F12.7,2X,5HRMSE=F12.7)
9     FORMAT(3X,8HPERCENT=F12.7)
11    FORMAT(I2,F12.7,I2)
13    FORMAT(I4,3F8.4)
14    FORMAT(2X,3HNO.,10X,3HMAE,10X,3H2SD,10X,3H0/0,10X,3HMSE,10X,
     1  3H2SD,10X,3H0/0,10X,4HRMSE)
15    FORMAT(2X,I2,6X,7F13.7)
16    FORMAT(20HSUBOPT. DEMOD. NO. 1)
17    FORMAT( 3HSN=F5.3,2X,4HRHO=F5.3,2X,15HNO. ITERATIONS=I4)
      END
```

Table A-3-4. Computer Program, Rauch's Demodulator

```
      DIMENSION XS(64),JC(64,6),NN(2,32),II(5),PS(32,6),PM(2),PSS(6)
     1W(32,5),XXS(64),WT(512),PE(6),PER(63),P(777)
C  BYPASSES UNDERFLOW ERROR,
      CALL FTRAP
C  COMPUTES W(I,K)=y_IK ÷ S/N FOR 5-BIT WORDS,
      DO 30 K=1,5
 30   W(1,K)=-1,
      DO 62 I=1,31
      DO 40 K=1,5
 40   W(I+1,K)=W(I,K)
      DO 50 K=1,5
      N=6-K
      W(I+1,N)=W(I+1,N)+1,
      IF(W(I+1,N)) 51,51,50
 51   W(I+1,N)= 1,
      GO TO 62
 50   W(I+1,N)=-1,
  62  CONTINUE
C
C  COMPUTES AND STORES P(I)= 1/√(2π) ∫_{-∞}^{I} e^(-x²/2) dx.
C
      B=SQRT (1./6.2832)*,01
      P(389)=,5
      P(390)=,496
      X=,015
      DO 400 I=391,777
      P(I)=P(I-1)-B*EXP (-,5*X**2)
      X=X+,01
 400  CONTINUE
      DO 441 I=1,388
      N=778-I
 441  P(I)=1,-P(N)
C  SETS CONSTANTS, XLIM= ACCURACY LIMIT, XS(I)=1/f_Y(I),
C  RH= ρ12 ,RHH= ρ13 ,XMY=m_Y, SY=σ_Y, MM=m, MM1=2^m,
C
      READ INPUT TAPE 7,401,XS
 405  READ INPUT TAPE 7,402,RH,RHH,XMY,SY,MM
      READ INPUT TAPE 7,950,XLIM
      MM1=2**MM
      MM2=MM1/2
      MM3=MM1-1
C  COMPUTES JC(I,K)= y_IK ÷ S/N.
      DO 70 K=1,MM
 70   JC(1,K)=-1
      DO 72 I=1,MM3
      DO 80 K=1,MM
```

Table A-3-4 (Continued)

```
 80    JC(I+1,K)=JC(I,K)
       DO 81 K=1,MM
       N=MM-K+1
       JC(I+1,N)=JC(I+1,N)+1
       IF(JC(I+1,N)) 82,82,81
 82    JC(I+1,N)= 1
       GO TO 72
 81    JC(I+1,N)=-1
 72    CONTINUE
C   SETS CONSTANTS FOR THE INTEGRATION.
       E=1.-2.*RH**2*(1.-RHH)-RHH**2
       F=1.-RH**2
       S=1./F
       SZ=SY**2
       YS=SZ*F
       SX=SZ*E*S
       S1=-.5/SZ
       S2=-.5/YS
       S3=-.5/SX
       F1=(1.-RH)*XMY
       F2=RH*(1.-RHH)*S
       F3=(RH**2-RHH)*S
       F4=XMY*(1.-F2+F3)
       EP3=XLIM
       EP2=EP3*10.
       EP1=EP2*10.
C   COMPUTES WT(MX)=fY(K,J,I) BY INTEGRATION OF fx (Z,Y,X).
C
C   CALC1 CALLS ON THE INTEGRATION SUBPROGRAM (AT END OF PROGRAM)
       DO 110 K=1,MM1
       XK=K
       Z=XK-1.-XMY
       DO 110 J=1,MM1
       XJ=J
       Y=XJ-1.-XMY
       DO 110 I=1,K
       XI=I
       X=XI-1.-XMY
       MX=K*64+J*8+I-72
       WT(MX)=   CALC1(X,Y,Z,S1,S2,S3,EP1,EP2,EP3,F1,F2,F3,F4,RH)
 110   CONTINUE
C   COMPLETES COMPUTATION OF WT(MX)=fY(I,J,K) AND PRINTS RESULTS.
       DO 111 I=1,MM3
       N=I+1
       DO 111 J=1,MM1
       DO 111 K=N,MM1
       MX=I*64+J*8+K-72
       MY=K*64+J*8+I-72
 111   WT(MX)=WT(MY)
       WRITE OUTPUT TAPE 6,101,WT
```

Table A-3-4 (Continued)

```
C   COMPUTES AND PRINTS PS(I,K)=PBS IK
      DO 470 K=1,MM
      N=1
      M=1
      DO 415 I=1,MM1
      IF(JC(I,K)) 412,413,413
  412 NN(1,N)=I
      N=N+1
      GO TO 415
  413 NN(2,M)=I
      M=M+1
  415 CONTINUE
      DO 450 I=1,16
      PS(I,K)=0.
      DO 416 KK=1,5
  416 II(KK)=.5*W(I,KK)+1.5
      DO 450 M1=1,MM2
      L=II(1)
      J1=NN(L,M1)
      DO 450 M2=1,MM2
      L=II(2)
      J2=NN(L,M2)
      DO 450 M3=1,MM2
      L=II(3)
      J3=NN(L,M3)
      MX=J1*64+J2*8+J3-72
      A=WT(MX)*XS(J3)
      DO 450 M4=1,MM2
      L=II(4)
      J4=NN(L,M4)
      DO 450 M5=1,MM2
      L=II(5)
      J5=NN(L,M5)
      MX=J3*64+J4*8+J5-72
      PS(I,K)=PS(I,K)+A*WT(MX)
  450 CONTINUE
      PSS(K)=0.
      DO 455 I=1,16
      NNN=33-I
      PS(NNN,K)=PS(I,K)
      PSS(K)=PSS(K)+PS(I,K)+PS(NNN,K)
  455 CONTINUE
      DO 460 I=1,32
  460 PS(I,K)=PS(I,K)/PSS(K)
      WRITE OUTPUT TAPE 6,101,(PS(I,K),I=1,32)
  470 CONTINUE
C   COMPUTES XXS(I)=f Y(I).
      ZZ=0.
```

Table A-3-4 (Continued)

```
      N=(ABSF(ZZ-XMY)-.5)/SY*100.+389.5
      XXS(1)=P(N)
      XXS(MM1)=XXS(1)
      XB=XXS(1)
      DO 830 I=2,MM2
      ZZ=I-1
      N=(ABSF(ZZ-XMY)-.5)/SY*100.+389.5
      J=MM1-I+1
      XXS(I)=P(N)-XB
      XXS(J)=XXS(I)
      XB=XB+XXS(I)
  830 CONTINUE
C  READS IN S/N AND SPACES PRINTER. A BLANK CARD RETURNS
C  THE PROGRAM TO STATEMENT 405.
  399 READ INPUT TAPE 7,401,SN
      IF(SN) 405,405,406
  406 CONTINUE
      WRITE OUTPUT TAPE 6,411
C  COMPUTES AND PRINTS WWW1= w1K, ETC.,THE BEST WEIGHTINGS
C
C  AND PE(K)= THE PROBABILITY OF AN ERROR IN THE KTH BIT.
      AB=SN*100.
      DO 500 K=1,MM
      PM(2)=100.
      W3=1.02
      DO 740 L=1,41
      W3=W3-.02
      W2=(1.-W3)*.5
      W1=0.
      N=(L+1)/2
      PM(1)=100.
      DO 700 LL=1,N
      SW=AB/SQRT (2.*(W1**2+W2**2)+W3**2)
      PP=0.
      DO 600 I=1,16
      X=    ((W(I,1)+W(I,5))*W1+ (W(I,2)+W(I,4))*W2+W(I,3)*W3)*SW+W(I,3)
      J=X+389.5
      IF(J) 611,611,612
  611 J=1
      GO TO 600
  612 IF(J-777) 600,600,621
  621 J=777
  600 PP=PP+PS(I,K)*P(J)
      PP=PP*2.
      IF(PP-PM(1)) 710,710,720
  710 PM(1)=PP
      WW3=W3
      WW2=W2
      WW1=W1
  720 W2=W2-.01
```

Table A-3-4 (Continued)

```
      W1=W1+.01
 700  CONTINUE
      IF(PM(1)-PM(2)) 730,730,740
 730  PM(2)=PM(1)
      WWW1=WW1
      WWW2=WW2
      WWW3=WW3
 740  CONTINUE
      PE(K)=PM(2)
 500  WRITE OUTPUT TAPE 6,401,WWW1,WWW2,WWW3
      WRITE OUTPUT TAPE 6,101,PE
C  COMPUTES AND PRINTS PER(J)= THE PROBABILITY OF AN ERROR OF J.
      DO 820 J=1,MM3
      PER(J)=0.
      M=MM1-J
      DO 820 I=1,M
      N=I+J
      PR=1.
      DO 810 KK=1,MM
      XJ=(XABSF(JC(I,KK)-JC(N,KK)))/2
 810  PR=(PE(KK)*XJ+(1.-XJ)*(1.-PE(KK)))*PR
      PER(J)=PER(J)+PR*(XXS(I)+XXS(N))
 820  CONTINUE
      WRITE OUTPUT TAPE 6,101,PER
C  COMPUTES AND PRINTS XMAE=MAE AND RMSE=RMSE, THEN RETURNS
C  TO READ A NEW S/N.
      XMAE=0.
      XMSE=0.
      DO 805 I=1,MM3
      XI=I
      XMAE=XMAE+XI*PER(I)
 805  XMSE=XMSE +PER(I)*XI**2
      RMSE=SQRT(XMSE)
      WRITE OUTPUT TAPE 6,3
      WRITE OUTPUT TAPE 6,2,SN,RH,RHH,XMAE,RMSE
      GO TO 399
 2    FORMAT(1X,F6.4,2X,F6.4,2X,F6.4,4X,F10.7,2X,F10.7)
 3    FORMAT(1X,3HS/N,5X,3HRHO,5X,5HRHO13,6X,3HMAE,9X,4HRMSE)
 5    FORMAT(1H1)
 101  FORMAT(8E14.7)
 401  FORMAT( 8F10.5)
 402  FORMAT(4F10.5,I2)
 404  FORMAT(10I10)
 411  FORMAT(1H0)
 950  FORMAT(5E14.7)
      END
```

Table A-3-4 (Continued)

```
C  THESE 3 SUBPROGRAMS EVALUATE THE TRIPPLE INTEGRAL
C  BY SIMPSON'S 1/3 RULE.
C  THE INTEGRATION IS DONE AS
C
C    CALC3(Y,Z)= ∫f_x|yz(X|Y, Z)dX,
C
C    CALC2(Z)= ∫f_y|z(Y|Z) CALC3 (Y, Z)dY,
C
C    CALC1=∫f_z(Z) CALC 2 (Z)dZ.
C
      FUNCTION  CALC1(X,Y,Z,S1,S2,S3,EP1,EP2,EP3,F1,F2,F3,F4,RH)
      A=Z-.5
      B=Z+.5
      C=        EXP (S1*A**2)*CALC2(X,Y,A,S2,S3,EP2,EP3,F1,F2,F3,F4,RH)
      D=        EXP (S1*B**2)*CALC2(X,Y,B,S2,S3,EP2,EP3,F1,F2,F3,F4,RH)
      SUM=C+D
      EPS=EP1
      PSUM4=0.
      DO 250 I=1,8
      H=1./(2.**I)
      L=2**I-1
      SUM2=PSUM4*.5
      TSUM4=0.
      DO 230 N=1,L,2
      ZN=N
      E= A+ZN*H
      E=        EXP (S1*E**2)*CALC2(X,Y,E,S2,S3,EP2,EP3,F1,F2,F3,F4,RH)
  230 TSUM4=TSUM4+4.*E
      PSUM4=PSUM4+TSUM4
      IF(I-1) 236,236,240
  236 Z1=(H/3.)*(SUM+SUM2+TSUM4)
      GO TO 250
  240 Z2=(H/3.)*(SUM+SUM2+TSUM4)
      DIFA=Z2-Z1
      IF(ABSF(DIFA)-EPS) 256,246,246
  246 Z1=Z2
  250 CONTINUE
  256 CONTINUE
      CALC1=Z2
      RETURN
      END

      FUNCTION                 CALC2(X,Y,Z,S2,S3,EP2,EP3,F1,F2,F3,F4,RH)
      A=Y-.5
      B=Y+.5
      XMZ=F1+RH*Z
      C=        EXP (S2*(A-XMZ)**2)*CALC3(X,A,Z,S3,EP3,F2,F3,F4)
      D=        EXP (S2*(B-XMZ)**2)*CALC3(X,B,Z,S3,EP3,F2,F3,F4)
```

Table A-3-4 (Continued)

```
      EPS=EP2
      SUM=C+D
      PSUM4=0.
      DO 250 I=1,8
      H=1./(2.**I)
      L=2**I-1
      SUM2=PSUM4*.5
      TSUM4=0.
      DO 230 N=1,L,2
      ZN=N
      E= A+ZN*H
      E=      EXP (S2*(E-XMZ)**2)*CALC3(X,E,Z,S3,EP3,F2,F3,F4)
230   TSUM4=TSUM4+4.*E
      PSUM4=PSUM4+TSUM4
      IF(I-1) 236,236,240
236   Z1=(H/3.)*(SUM+SUM2+TSUM4)
      GO TO 250
240   Z2=(H/3.)*(SUM+SUM2+TSUM4)
      DIFA=Z2-Z1
      IF(ABSF(DIFA)-EPS) 256,246,246
246   Z1=Z2
250   CONTINUE
256   CONTINUE
      CALC2=Z2
      RETURN
      END


      FUNCTION                         CALC3(X,Y,Z,S3,EP3,F2,F3,F4)
      XMYZ=F4-F3*Z+F2*Y
      A=X-.5
      B=X+.5
      C=      EXP (S3*(A-XMYZ)**2)
      D=      EXP (S3*(B-XMYZ)**2)
      EPS=EP3
      SUM=C+D
      PSUM4=0.
      DO 250 I=1,8
      H=1./(2.**I)
      L=2**I-1
      SUM2=PSUM4*.5
      TSUM4=0.
      DO 230 N=1,L,2
      ZN=N
      E= A+ZN*H
      E=      EXP (S3*(E-XMYZ)**2)
230   TSUM4=TSUM4+4.*E
      PSUM4=PSUM4+TSUM4
      IF(I-1) 236,236,240
```

Table A-3-4 (Continued)

```
236   Z1=(H/3.)*(SUM+SUM2+TSUM4)
      GO TO 250
240   Z2=(H/3.)*(SUM+SUM2+TSUM4)
      DIFA=Z2-Z1
      IF(ABSF(DIFA)-EPS) 256,246,246
246   Z1=Z2
250   CONTINUE
256   CONTINUE
      CALC3=Z2
      RETURN
      END
```

Table A-3-5. Computer Program, Suboptimal Number 3

```
      DIMENSION JC(64,6),CU(5,6),CV(6)    ,JCD(6),Z(5),VS(5,6),JYT(5)
      DIMENSION W1(6),W2(6),W3(6),P(64),XS2(64)
C  BYPASSES UNDERFLOW ERROR,
      CALL FTRAP
C  SETS CONSTANTS, MM=m,MM1=2^m,XMY=m_Y,SY=σ_Y,
      RNO=0,
      RN1=0,
 400  READ INPUT TAPE 7,11,MM,AA
      MM1=2**MM
      XMM1=MM1
      XMM=MM
      MM2=MM1-1
      MM3=MM+1
      XMM2=MM2
      XMY=XMM2/2.
      SY=XMY/AA
      RSY=1./SY
C  COMPUTES JC(I,K)=y_IK ÷ S/N.
      DO 30 K=1,MM
  30  JC(1,K)=-1
      DO 62 I=1,MM2
      DO 40 K=1,MM
  40  JC(I+1,K)=JC(I,K)
      DO 50 K=1,MM
      N=MM-K+1
      JC(I+1,N)=JC(I+1,N)+1
      IF(JC(I+1,N)) 51,51,50
  51  JC(I+1,N)=1
      GO TO 62
  50  JC(I+1,N)=-1
  62  CONTINUE
C  COMPUTES XS2(I)=1/f_Y(I),
      DO 63 I=1,MM1
      Y=I-1
  63  XS2(I)=.5*(((Y-XMY)*RSY)**2)
C  READS IN INPUT DATA AND SETS CONSTANTS, SN=S/N,LL=NO, OF
C  ITERATIONS,RH= ρ12  ,RHH= ρ13  ,W1(I)=w_1I , ETC,,PER= WEIGHTING
C
C  FACTOR, SYM AND SYN ARE THE σ's FOR THE x DISTRIBUTION,
  70  READ INPUT TAPE 7,13,LL,SN,RH,RHH
      IF (LL) 400,400,701
 701  CONTINUE
      READ INPUT TAPE 7,23,W1,W2,W3
      READ INPUT TAPE 7,10,PER
      XLL=LL
      F=RH**2
      E=1.-F
      SYM=SY*SQRT (E)
```

Table A-3-5 (Continued)

```
      E=1./E
      C=(F-RHH)*E
      B=RH*(1.-RHH)*E
      A=(1.-P+C)*XMY
      D=(1.-RH)*XMY
      SYN=SY*SQRT ((1.-2.*F*(1.-RHH)-RHH**2)*E)
      T=RSY**2*.5*E
      ER=0.
      ERR=0.
      ERS=0.
      VER=0.
      VERS=0.
      VERSS=0.
C  INITIATES LL ITERATIONS
      DO 500 L=1,LL
C  GENERATES THE NEEDED GAUSSIAN RANDOM NUMBERS.
      DO 600 I=1,5
      Z(I)=RANDND(0.,1.,RN0)
      DO 600 J=1,MM
 600  VS(I,J)=RANDND(0.,1.,RN1)
C  COMPUTES X1=x1 , ETC.,XN= mx.
C
      X3=SY*Z(3)+XMY
      XM=D+RH*X3
      X2=SYM*Z(2)+XM
      X4=SYM*Z(4)+XM
      XN=A+B*X2-C*X3
      X1=SYN*Z(1)+XN
      XN=A+B*X4-C*X3
      X5=SYN*Z(5)+XN
C  COMPUTES JYT(I)=YI  AND CU(I,K)=zIK.
      JYT(1)=X1+.5
      JYT(2)=X2+.5
      JYT(3)=X3+.5
      JYT(4)=X4+.5
      JYT(5)=X5+.5
      DO 140 I=1,5
      IF (JYT(I)) 99,130,300
 99   JYT(I)=0
      GO TO 130
 300  IF (JYT(I)-MM2) 130,130,110
 110  JYT(I)=MM2
 130  J=JYT(I)+1
      DO 140 K=1,MM
      US=VS(I,K)
      IF(JC(J,K)) 136,136,138
 136  CU(I,K)=US-SN
      GO TO 140
 138  CU(I,K)=US+SN
 140   CONTINUE
```

Table A-3-5 (Continued)

```
C   COMPUTES P(T)= h(I|z(t))          CV(K)=H_jK .
      DO 207 K=1,MM
      CV(K)=W1(K)*(CU(1,K)+CU(5,K))+W2(K)*(CU(2,K)+CU(4,K))+W3(K)*CU(3,K
     1)
  207 CONTINUE
      Y3=JYT(3)
      PS=0.
      DO 220 I=1,MM1
      U=0.
      DO 218 K=1,MM
      IF(JC(I,K)) 214,214,216
  214 U=U-CV(K)
      GO TO 218
  216 U=U+CV(K)
  218 CONTINUE
      V=U*SN*PER-XS2(I)
      P(I)=EXP (V)
  220 PS=PS+P(I)
      DO 230 I=1,MM1
  230 P(I)=P(I)/PS
C   COMPUTES YH= Ŷ_j.
      YH=0.
      DO 240 I=1,MM1
      D2=I-1
  240 YH=YH+D2*P(I)
C
C   COMPUTES AND STORES VER= Σ_{k=1}^{L} (AE_1)_k   ,VERS= Σ_{k=1}^{L} (SE_1)_k  ,VERSS= Σ_{k=1}^{L} (SE_1)_k^2
C
      ER1=ABSF(Y3-YH)
      ER2=ER1**2
      VER=VER+ER1
      VERS=VERS+ER2
      VERSS=VERSS+ER2**2
C   STOPS ITERATIONS AND SPACES PRINTER.
  500 CONTINUE
      WRITE OUTPUT TAPE 6,5
C   COMPUTES AND PRINTS VER=MÂE,VSD1=2σ̂_MÂE,PC31= % FOR 2σ̂_MÂE
C
C   VERS=MŜE, VSD2=2σ̂_MŜE,PC32= % FOR 2σ̂_MŜE, RMSE=RMŜE,
C
      VER=VER/XLL
      VERS=VERS/XLL
      VSD1=2.*SQRT ((VERS-VER**2)/XLL)
      VSD2=2.*SQRT ((VERSS/XLL-VERS**2)/XLL)
      RMSE=SQRT (VERS)
      PC31=VSD1/VER*100.
      PC32=VSD2/VERS*100.
      WRITE OUTPUT TAPE 6,3
      WRITE OUTPUT TAPE 6,3
      WRITE OUTPUT TAPE 6,21
      WRITE OUTPUT TAPE 6,6,MM
      WRITE OUTPUT TAPE 6,2,SN,RH,RHH,LL
```

Table A-3-5 (Continued)

```
      WRITE OUTPUT TAPE 6,22,PER
      WRITE OUTPUT TAPE 6,24,W1,W2,W3
      WRITE OUTPUT TAPE 6,3
      WRITE OUTPUT TAPE 6,14
      WRITE OUTPUT TAPE 6,15,  VER,VSD1,PC31,VERS,VSD2,PC32,RMSE
C   RETURNS TO READ ANOTHER SET OF DATA.
      GO TO 70
 1    FORMAT(6I6)
 2    FORMAT(3HSN=F5.3,2X,4HRHO=F5.3,2X,6HRHO13=F5.3,,2X,
     1   15HNO. ITERATIONS=I4)
 3    FORMAT(1H0)
 4    FORMAT(8E14.6)
 5    FORMAT(1H1)
 6    FORMAT(2X,I2,1X,9HBIT WORDS)
 7    FORMAT(28HSUBOPTIMAL DEMODULATOR NO. 2)
 8    FORMAT(2X,I2,4X,4HMSE=F12.7,2X,4H2SD=F12.7,2X,5HRMSE=F12.7)
 9    FORMAT(3X,8HPERCENT=F12.7)
 10   FORMAT(10F10.4)
 11   FORMAT(I2,F12.7,I2)
 13   FORMAT(I4,3F8.4)
 14   FORMAT(          10X,3HMAE,10X,3H2SD,10X,3H0/0,10X,3HMSE,10X,
     1  3H2SD,10X,3H0/0,10X,4HRMSE)
 15   FORMAT(5X,7F13.7)
 16   FORMAT(20HSUBOPT. DEMOD. NO. 1)
 17   FORMAT( 3HSN=F5.3,2X,4HRHO=F5.3,2X,15HNO. ITERATIONS=I4)
 20   FORMAT(8F10.4)
 21   FORMAT(28HSUBOPTIMAL DEMODULATOR NO. 3)
 22   FORMAT(3X,10HWEIGHTING=F10.6)
 23   FORMAT(6F2.2)
 24   FORMAT(6(3X,F3.2))
 25   FORMAT(15HEXECUTION TIME=I4,9HMINUTES, F4.1,7HSECONDS)
      END
```

Appendix IV

COMPUTATION RESULTS

Tables A-4-1 through A-4-7 give the computational results for the optimal and suboptimal demodulators. The following symbols, although used in the main body of this dissertation, are repeated here for convenience.

- n: number of PCM words considered at any one time in the demodulation process.
- m: the number of bits in each PCM word.
- S/N: the RMS signal-to-noise ratio.
- ρ_{12}: the correlation coefficient between adjacent data samples, x_i^* and x_{i+1}^*.
- ρ_{13}: the correlation coefficient between alternating data samples, x_i^* and x_{i+2}^*.
- N: the number of iterations used in the Monte Carlo simulation.
- A: the standard deviation of the modified noise distribution where importance sampling was used.
- B: the mean of the modified noise distribution for importance sampling.
- w_{ir}: the weighting factor for the rth bit of the ith word in Rauch's suboptimal demodulator.
- WT: the weighting used in suboptimal demodulator number 3.

All normalized (abbreviated as "Nor." in the tables) values are normalized by dividing by the peak-to-peak signal.

Table A-4-8 gives a summary of the computation times required on the University of Michigan IBM 7090 Computer to generate the data in Tables A-4-1 through A-4-7.

TABLE A-4-1

Data, Present Day Demodulators (Bit-by-bit Correlation)

n	m	S/N	NOR. MAE	NOR. RMSE
1	3	.707	.1976	.2986
1	3	1.0	.1400	.2491
1	3	1.414	.07403	.1796
1	3	2.0	.02231	.09802
1	3	2.83	.002296	.03138
1	6	.707	.1716	.2550
1	6	1.0	.1231	.2130
1	6	1.414	.06600	.1537
1	6	2.0	.02000	.0838
1	6	2.83	.002056	.02684
1	6	3.55	.0001081	.007920

TABLE A-4-2

Data, Minimum P_e Demodulator

a. Performance.

n	m	S/N	ρ_{12}	N	A	B	Nor. $\widehat{MAE}$	Nor. $2\hat{\sigma}_{\widehat{MAE}}$	Nor. $\widehat{RMSE}$	Nor. $2\hat{\sigma}_{\widehat{RMSE}}$
1	3	.707		300			.12032	.006666	.20477	.02253
1	3	1.0		400			.095235	.00673	.18802	.02274
1	3	1.414		500			.048052	.005256	.13810	.01834
1	3	2.0		750	1.25	.5	.016348	.002045	.082106	.006910
1	3	2.83		3000	1.25	.75	.0018529	.0002226	.027894	.002165
2	3	.707	.9	150			.10418	.00943	.18522	.01228
2	3	1.0	.9	300			.071479	.007455	.15807	.01194
2	3	1.414	.9	750			.028571	.003181	.095817	.007529
2	3	2.0	.9	1700	1.25	.3	.0058004	.0006871	.041351	.003111
2	3	.707	.95	200			.10327	.00997	.19371	.01093
2	3	1.0	.95	350			.061411	.006164	.14832	.01050
2	3	1.414	.95	1000			.026715	.002795	.096613	.007202
2	3	2.0	.95	2100	1.25	.3	.0049600	.0006420	.036658	.003139
2	3	.707	.98	200			.093823	.008998	.18452	.01197
2	3	1.0	.98	400			.061052	.006524	.15221	.01094
2	3	1.414	.98	1200			.020332	.002194	.084825	.006279
2	3	2.0	.98	2400	1.25	.2	.0047625	.0006602	.035863	.002675
3	3	.707	.95	200			.084192	.009140	.17037	.01357
3	3	1.0	.95	400			.041154	.005169	.11487	.01085
3	3	1.414	.95	1200			.012987	.001630	.058529	.005749
3	3	2.0	.95	4000	1.25	.2	.0024691	.0003162	.021839	.001452
3	3	.707	.98	200			.072387	.008436	.15843	.01255
3	3	1.0	.98	400			.036790	.005083	.11262	.01047

TABLE A-4-2 (Continued)

n	m	S/N	ρ_{12}	N	A	B	Nor. $\hat{MAE}$	Nor. $2\hat{\sigma}_{\hat{MAE}}$	Nor. $\hat{RMSE}$	Nor. $2\hat{\sigma}_{\hat{RMSE}}$
3	3	1.414	.98	1200			.0098540	.001598	.055510	.007258
3	3	2.0	.98	4000	1.25	.2	.0018107	.0002899	.018405	.001413
1	6	.707		100			.11750	.01093	.18246	.01297
1	6	1.0		200			.090210	.008360	.16734	.01118
1	6	1.414		400			.048910	.005164	.11963	.00895
1	6	2.0		1200	1.25	.3	.017388	.001660	.074620	.004923
1	6	2.83		2200	1.25	.5	.0017940	.0002746	.023380	.002383
2	6	.707	.95	200			.09691	.007470	.16167	.01033
2	6	1.0	.95	350			.064856	.005635	.13166	.00898
2	6	1.414	.95	1000			.026294	.002212	.077552	.005656
2	6	2.0	.95	2000	1.25	.1	.0057082	.0006602	.031325	.003589
2	6	2.83	.95	1500	1.25	.2	.00038188	.0008398	.0059953	.0008899
2	6	.707	.98	200			.094678	.006803	.15704	.00911
2	6	1.0	.98	400			.059242	.005443	.12705	.00898
2	6	1.414	.98	1200			.022658	.002021	.075104	.005660
2	6	2.0	.98	2400	1.25	.1	.0045210	.0005698	.030870	.004121
2	6	2.83	.98	1500	1.25	.2	.00030240	.00007326	.0053121	.0012047
2	6	.707	.995	200			.090361	.007634	.15662	.009874
2	6	1.0	.995	500			.058117	.004622	.129833	.007534
2	6	1.414	.995	1500			.0208776	.001772	.075956	.004950
2	6	2.0	.995	2700	1.25	.1	.0027163	.0003459	.023376	.003219
2	6	2.83	.995	1500	1.25	.2	.00014613	.00003809	.0040252	.0013996

b. Bias.

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	.707	.95	50	0	23.9800	2.8552
2	6	.707	.95	50	8	25.5600	3.4008

TABLE A-4-2 (Continued)

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	.707	.95	50	16	22.4000	2.2915
2	6	.707	.95	50	24	25.7200	1.6532
2	6	.707	.95	50	31	30.1000	2.6200
2	6	1.0	.95	50	0	17.8000	3.3856
2	6	1.0	.95	50	8	16.5800	3.1262
2	6	1.0	.95	50	16	20.3800	2.0991
2	6	1.0	.95	50	24	24.7200	1.3182
2	6	1.0	.95	50	31	28.6400	1.8597
2	6	1.414	.95	50	0	6.1400	2.6272
2	6	1.414	.95	50	8	11.1600	1.8833
2	6	1.414	.95	50	16	17.6600	1.6929
2	6	1.414	.95	50	24	24.5800	1.4914
2	6	1.414	.95	50	31	30.4200	0.9158
2	6	.707	.995	50	0	24.4000	2.9864
2	6	.707	.995	50	8	23.7800	3.0887
2	6	.707	.995	50	16	21.6400	2.4519
2	6	.707	.995	50	24	24.5000	1.2703
2	6	.707	.995	50	31	30.2600	1.7878
2	6	1.0	.995	50	0	16.6800	3.8890
2	6	1.0	.995	50	8	16.3200	2.9947
2	6	1.0	.995	50	16	21.7400	2.8932
2	6	1.0	.995	50	24	24.6200	1.0596
2	6	1.0	.995	50	31	29.7000	0.8080
2	6	1.414	.995	50	0	5.7600	3.0324
2	6	1.414	.995	50	8	10.1200	1.7496
2	6	1.414	.995	50	16	17.2800	1.4154
2	6	1.414	.995	50	24	24.0600	0.4433
2	6	1.414	.995	50	31	30.5800	0.2889

TABLE A-4-2 (Continued)

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	3	.707	.9	300	0	2.7733	.1567
2	3	.707	.9	300	1	2.7300	.1632
2	3	.707	.9	200	2	2.7200	.1457
2	3	.707	.9	200	3	3.0700	.1221
2	3	1.0	.9	300	0	1.85000	.1833
2	3	1.0	.9	300	1	1.8233	.1526
2	3	1.0	.9	200	2	2.3600	.1233
2	3	1.0	.9	200	3	3.0400	.1173
2	3	1.414	.9	300	0	.8333	.1537
2	3	1.414	.9	300	1	1.3167	.1089
2	3	1.414	.9	200	2	2.1100	.0967
2	3	1.414	.9	200	3	2.9700	.0706
2	3	.707	.95	300	0	2.6000	.1655
2	3	.707	.95	300	1	2.6867	.1765
2	3	.707	.95	200	2	2.6550	.1446
2	3	.707	.95	200	3	3.0450	.1143
2	3	1.0	.95	300	0	1.8033	.1875
2	3	1.0	.95	300	1	1.8333	.1508
2	3	1.0	.95	200	2	2.2650	.1081
2	3	1.0	.95	200	3	2.9100	.0874
2	3	1.414	.95	300	0	.5600	.1293
2	3	1.414	.95	300	1	1.2733	.1023
2	3	1.414	.95	200	2	2.1150	.0710
2	3	1.414	.95	200	3	2.9650	.0409

TABLE A-4-2 (Continued)

c. Robustness

n	m	Dem.[1] S/N	Act.[2] S/N	Dem. ρ_{12}	Act. ρ_{12}	N	A	B	Nor. $\hat{MAE}$	Nor. $2\hat{\sigma}_{\hat{MAE}}$	Nor. $\hat{RMSE}$	Nor. $2\hat{\sigma}_{\hat{RMSE}}$
2	3	1.0	.707	.95	.95	700			.10490	.01228	.19333	.01709
2	3	1.0	.841	.95	.95	1200			.089762	.008941	.17900	.01309
2	3	1.0	1.19	.95	.95	2400			.046964	.004746	.12539	.009406
2	3	1.0	1.414	.95	.95	3000			.031095	.003534	.10165	.00850
2	3	1.0	1.0	.95	.9	1200			.072381	.007975	.15595	.01262
2	3	1.0	1.0	.95	.98	1600			.060625	.006602	.14529	.01140
2	3	2.0	1.414	.95	.95	4000			.025893	.002862	.094139	.007470
2	3	2.0	1.68	.95	.95	3400	1.25	.2	.012904	.001467	.062185	.004735
2	3	2.0	2.37	.95	.95	10000	1.25	.3	.0020306	.0003334	.021611	.002151
2	3	2.0	2.83	.95	.95	12000	1.4	.3	.00066191	.0001881	.011167	.001579
2	3	2.0	2.0	.95	.9	6000	1.25	.3	.010311	.001289	.053326	.004122
2	3	2.0	2.0	.95	.98	6000	1.25	.3	.0037823	.0004731	.032363	.002455

[1] Demodulator value.

[2] Actual value.

TABLE A-4-3

Data, Minimum MAE Demodulator

a. Performance

n	m	S/N	ρ_{12}	N	A	B	Nor. $\hat{MAE}$	Nor. $2\hat{\sigma}_{\hat{MAE}}$	Nor. $\hat{RMSE}$	Nor. $2\hat{\sigma}_{\hat{RMSE}}$
1	3	.707		300			.11612	.00584	.19028	.01515
1	3	1.0		400			.092388	.006200	.17614	.01725
1	3	1.414		500			.047589	.005125	.13360	.01612
1	3	2.0		750	1.25	.5	.016314	.002045	.081469	.006932
1	3	2.83		3000	1.25	.75	.0018515	.0002226	.027829	.002169
2	3	.707	.9	150			.10200	.00879	.17615	.01010
2	3	1.0	.9	300			.070100	.007174	.14980	.01056
2	3	1.414	.9	750			.028380	.003130	.093512	.007095
2	3	2.0	.9	1700	1.25	.3	.0057951	.0006868	.041234	.003112
2	3	.707	.95	200			.098849	.008985	.17947	.01024
2	3	1.0	.95	350			.060585	.005942	.14377	.00960
2	3	1.414	.95	1000			.026551	.002753	.095310	.007060
2	3	2.0	.95	2100	1.25	.3	.0049501	.0006385	.036429	.003005
2	3	.707	.98	200			.092002	.008542	.17733	.01072
2	3	1.0	.98	400			.060195	.006331	.14605	.00996
2	3	1.414	.98	1200			.020208	.002167	.083759	.006159
2	3	2.0	.98	2400	1.25	.2	.0047442	.0006548	.035675	.002660
3	3	.707	.95	200			.082247	.008637	.16330	.01269
3	3	1.0	.95	400			.040618	.005016	.11172	.01031
3	3	1.414	.95	1200			.012938	.001617	.057907	.005704
3	3	2.0	.95	4000	1.25	.2	.0024688	.0004354	.019243	.001195
3	3	.707	.98	200			.071397	.008145	.15311	.01138
3	3	1.0	.98	400			.036332	.004903	.10878	.00925

TABLE A-4-3 (Continued)

n	m	S/N	ρ_{12}	N	A	B	Nor. $\widehat{MAE}$	Nor. $2\hat{\sigma}_{\widehat{MAE}}$	Nor. $\widehat{RMSE}$	Nor. $2\hat{\sigma}_{\widehat{RMSE}}$
3	3	1.414	.98	1200			.0098238	.001591	.055357	.007264
3	3	2.0	.98	4000	1.25	.2	.0018099	.0002899	.018369	.001413
1	6	.707		100			.10988	.00913	.16358	.00864
1	6	1.0		200			.086010	.007376	.15391	.00891
1	6	1.414		400			.047880	.004951	.11335	.00785
1	6	2.0		1200	1.25	.3	.017306	.001658	.073865	.004937
1	6	2.83		2200	1.25	.5	.0017930	.0002747	.02336	.002383
2	6	.707	.95	200			.092390	.006625	.14660	.00816
2	6	1.0	.95	350			.062300	.005234	.12195	.00790
2	6	1.414	.95	1000			.025953	.002169	.074632	.005202
2	6	2.0	.95	2000	1.25	.1	.0056768	.0006557	.030932	.003553
2	6	2.83	.95	1500	1.25	.2	.00038168	.00008397	.0059897	.0008908
2	6	.707	.98	200			.090003	.006177	.14438	.00712
2	6	1.0	.98	400			.056570	.004944	.11710	.00753
2	6	1.414	.98	1200			.022372	.001987	.072862	.005332
2	6	2.0	.98	2400	1.25	.1	.0045059	.0005691	.030681	.004130
2	6	2.83	.98	1500	1.25	.2	.00030237	.00007325	.0053025	.0012067
2	6	.707	.995	200			.086017	.006954	.14323	.00835
2	6	1.0	.995	500			.056605	.004417	.12400	.00693
2	6	1.414	.995	1500			.020675	.001750	.074705	.004836
2	6	2.0	.995	2700	1.25	.1	.0027113	.0003458	.023322	.003225
2	6	2.83	.995	1500	1.25	.2	.0014612	.00003809	.0040250	.0013997

b. Bias.

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	.707	.95	50	0	24.9400	2.5433

TABLE A-4-3 (Continued)

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	.707	.95	50	8	24.9800	2.8389
2	6	.707	.95	50	16	23.3800	1.8286
2	6	.707	.95	50	24	26.2400	1.4887
2	6	.707	.95	50	31	30.1000	1.9492
2	6	1.0	.95	50	0	17.9800	3.1972
2	6	1.0	.95	50	8	19.2400	3.0361
2	6	1.0	.95	50	16	20.1400	1.6415
2	6	1.0	.95	50	24	25.2400	1.2504
2	6	1.0	.95	50	31	28.6200	1.7563
2	6	1.414	.95	50	0	6.3600	2.6112
2	6	1.414	.95	50	8	11.1200	1.7921
2	6	1.414	.95	50	16	17.6000	1.6685
2	6	1.414	.95	50	24	24.4800	1.1103
2	6	1.414	.95	50	31	30.4200	0.8783
2	6	.707	.995	50	0	24.1400	2.5355
2	6	.707	.995	50	8	23.6800	2.7779
2	6	.707	.995	50	16	22.3800	1.8233
2	6	.707	.995	50	24	24.8000	1.0568
2	6	.707	.995	50	31	29.4000	1.4088
2	6	1.0	.995	50	0	15.8800	3.4915
2	6	1.0	.995	50	8	17.2800	2.9027
2	6	1.0	.995	50	16	21.8000	2.7966
2	6	1.0	.995	50	24	24.8600	1.1593
2	6	1.0	.995	50	31	29.5400	1.0388
2	6	1.414	.995	50	0	6.1000	2.9967
2	6	1.414	.995	50	8	9.8400	1.5863
2	6	1.414	.995	50	16	17.2800	1.4153
2	6	1.414	.995	50	24	24.1400	0.4525

TABLE A-4-3 (Continued)

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	1.414	.995	50	31	30.6000	0.2771
2	3	.707	.9	300	0	2.7567	.1345
2	3	.707	.9	300	1	2.7467	.1374
2	3	.707	.9	200	2	2.7450	.1393
2	3	.707	.9	200	3	3.0650	.1087
2	3	1.0	.9	300	0	1.8633	.1691
2	3	1.0	.9	300	1	1.9233	.1462
2	3	1.0	.9	200	2	2.3650	.1210
2	3	1.0	.9	200	3	3.0650	.1030
2	3	1.414	.9	300	0	.8267	.1489
2	3	1.414	.9	300	1	1.3233	.1062
2	3	1.414	.9	200	2	2.1100	.0967
2	3	1.414	.9	200	3	2.9650	.0669
2	3	.707	.95	300	0	2.5700	.1460
2	3	.707	.95	300	1	2.6933	.1532
2	3	.707	.95	200	2	2.6900	.1284
2	3	.707	.95	200	3	3.0350	.1004
2	3	1.0	.95	300	0	1.8200	.1731
2	3	1.0	.95	300	1	1.8567	.1397
2	3	1.0	.95	200	2	2.2550	.0860
2	3	1.0	.95	200	3	2.9300	.0866
2	3	1.414	.95	300	0	.5667	.1292
2	3	1.414	.95	300	1	1.2733	.0997
2	3	1.414	.95	200	2	2.1150	.0695
2	3	1.414	.95	200	3	2.9550	.0355

TABLE A-4-3 (Continued

c. Robustness

n	m	Dem.[1] S/N	Act.[2] S/N	Dem. ρ_{12}	Act. ρ_{12}	N	A	B	Nor. $\widehat{\text{MAE}}$	$2\hat{\sigma}$ Nor. $\widehat{\text{MAE}}$	Nor. $\widehat{\text{RMSE}}$	$2\hat{\sigma}$ Nor. $\widehat{\text{RMSE}}$
2	3	1.0	.707	.95	.95	700			.10163	.01160	.18406	.01650
2	3	1.0	.841	.95	.95	1200			.089286	.008562	.17311	.01260
2	3	1.0	1.19	.95	.95	2400			.045357	.004521	.11967	.00903
2	3	1.0	1.414	.95	.95	3000			.031190	.002813	.089255	.005971
2	3	1.0	1.0	.95	.9	1200			.070952	.007582	.14926	.01206
2	3	1.0	1.0	.95	.98	1600			.059018	.006263	.13846	.01103
2	3	2.0	1.414	.95	.95	4000			.025679	.002810	.092499	.007352
2	3	2.0	1.68	.95	.95	3400	1.25	.2	.012871	.001440	.061269	.004639
2	3	2.0	2.37	.95	.95	10000	1.25	.3	.0020192	.0003318	.021476	.002148
2	3	2.0	2.83	.95	.95	12000	1.4	.3	.00066198	.0001883	.011650	.001645
2	3	2.0	2.0	.95	.9	6000	1.25	.3	.010188	.001276	.052506	.004086
2	3	2.0	2.0	.95	.98	6000	1.25	.3	.0037833	.0004746	.032305	.002459

[1] Demodulator value.
[2] Actual value.

TABLE A-4-4

Data, Minimum MSE Demodulator

a. Performance

n	m	S/N	ρ_{12}	N	A	B	Nor. $\widehat{MAE}$	Nor. $2\hat{\sigma}_{\widehat{MAE}}$	Nor. $\widehat{RMSE}$	Nor. $2\hat{\sigma}_{\widehat{RMSE}}$
1	3	.707		300			.12880	.005750	.17861	.01190
1	3	1.0		400			.10943	.006490	.16162	.01333
1	3	1.414		500			.064360	.005993	.12007	.01158
1	3	2.0		750	1.25	.5	.023477	.002622	.071158	.005160
1	3	2.83		3000	1.25	.75	.0028050	.0003074	.024288	.001703
2	3	.707	.9	150			.11657	.00895	.16211	.00854
2	3	1.0	.9	300			.085167	.007748	.13484	.008667
2	3	1.414	.9	750			.038312	.003727	.082880	.005622
2	3	2.0	.9	1700	1.25	.3	.0085963	.0009338	.036136	.002500
2	3	.707	.95	200			.11382	.00917	.16426	.00825
2	3	1.0	.95	350			.077934	.006850	.12928	.00778
2	3	1.414	.95	1000			.036511	.003353	.083131	.005360
2	3	2.0	.95	2100	1.25	.3	.0072516	.0008310	.031708	.002280
2	3	.707	.98	200			.10864	.00912	.16171	.00838
2	3	1.0	.98	400			.075387	.007055	.13109	.00777
2	3	1.414	.98	1200			.028529	.002786	.074317	.004873
2	3	2.0	.98	2400	1.25	.2	.0067462	.0008337	.030766	.002071
3	3	.707	.95	200			.097411	.008598	.14596	.00938
3	3	1.0	.95	400			.051986	.005723	.098910	.007869
3	3	1.414	.95	1200			.018092	.001992	.050399	.004345
3	3	2.0	.95	4000	1.25	.2	.0037246	.0004354	.019243	.001195
3	3	.707	.98	200			.086631	.008823	.13936	.00927
3	3	1.0	.98	400			.048144	.005693	.098359	.007681
3	3	1.414	.98	1200			.014049	.002002	.047563	.005301

TABLE A-4-4 (Continued)

n	m	S/N	ρ_{12}	N	A	B	Nor. $\hat{MAE}$	Nor. $2\hat{\sigma}_{\hat{MAE}}$	Nor. $\hat{RMSE}$	Nor. $2\hat{\sigma}_{\hat{RMSE}}$
3	3	2.0	.98	4000	1.25	.2	.0027035	.0003897	.016127	.001134
1	6	.707		100			.11523	.00922	.15853	.00796
1	6	1.0		200			.096507	.007888	.14272	.00729
1	6	1.414		400			.057980	.005418	.10326	.00643
1	6	2.0		1200	1.25	.3	.024545	.002306	.065330	.004045
1	6	2.83		2200	1.25	.5	.0027085	.0003798	.020460	.001895
2	6	.707	.95	200			.09837	.006883	.13957	.006646
2	6	1.0	.95	350			.070526	.005734	.11189	.006606
2	6	1.414	.95	1000			.032569	.002501	.067032	.004228
2	6	2.0	.95	2000	1.25	.1	.008008	.0008693	.027238	.002699
2	6	2.83	.95	1500	1.25	.2	.00059449	.00012608	.0053779	.0007751
2	6	.707	.98	200			.096544	.006450	.13783	.006405
2	6	1.0	.98	400			.06469	.005366	.10781	.006253
2	6	1.414	.98	1200			.028638	.002358	.064732	.004197
2	6	2.0	.98	2400	1.25	.1	.0064309	.0007392	.026452	.003119
2	6	2.83	.98	1500	1.25	.2	.00047297	.00010796	.0047502	.0010369
2	6	.707	.995	200			.092539	.007174	.13626	.00691
2	6	1.0	.995	500			.066752	.005086	.11266	.005682
2	6	1.414	.995	1500			.027445	.002189	.066330	.003843
2	6	2.0	.995	2700	1.25	.1	.0039767	.0004643	.020427	.002514
2	6	2.83	.995	1500	1.25	.2	.00021765	.00005165	.0034731	.0012028

b. Bias

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	.707	.95	50	0	24.4108	1.8086
2	6	.707	.95	50	8	24.5309	2.2441

TABLE A-4-4 (Continued)

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	6	.707	.95	50	16	26.7010	1.4855
2	6	.707	.95	50	24	26.7010	1.6262
2	6	1.0	.95	50	0	17.7657	2.3623
2	6	1.0	.95	50	8	19.0609	2.3996
2	6	1.0	.95	50	16	21.2783	1.3350
2	6	1.0	.95	50	24	25.2728	1.1813
2	6	1.0	.95	50	31	28.7965	1.5390
2	6	1.414	.95	50	0	7.3629	2.3961
2	6	1.414	.95	50	8	12.343	1.4816
2	6	1.414	.95	50	16	18.6398	1.4836
2	6	1.414	.95	50	24	24.6507	1.0029
2	6	1.414	.95	50	31	29.6430	0.7953
2	6	.707	.995	50	0	23.8275	1.7939
2	6	.707	.995	50	8	23.5629	2.1150
2	6	.707	.995	50	16	23.8402	1.5933
2	6	.707	.995	50	24	25.8794	1.1289
2	6	.707	.995	50	31	29.4205	1.4088
2	6	1.0	.995	50	0	15.6348	2.6080
2	6	1.0	.995	50	8	17.6501	2.2808
2	6	1.0	.995	50	16	22.1645	2.1339
2	6	1.0	.995	50	24	25.1448	1.1671
2	6	1.0	.995	50	31	29.2906	0.8080
2	6	1.414	.995	50	0	6.7834	2.5404
2	6	1.414	.995	50	8	10.3441	1.4308
2	6	1.414	.995	50	16	17.9971	1.3357
2	6	1.414	.995	50	24	24.2682	0.3554
2	6	1.414	.995	50	31	30.3970	0.3505
2	3	.707	.9	300	0	2.6900	.0974

TABLE A-4-4 (Continued)

n	m	S/N	ρ_{12}	N	Y_2	$\hat{E}(\hat{Y}_2)$	$2\hat{\sigma}$
2	3	.707	.9	300	1	2.7588	.0976
2	3	.707	.9	200	2	2.8567	.1037
2	3	.707	.9	200	3	3.1407	.0903
2	3	1.0	.9	300	0	1.8947	.1225
2	3	1.0	.9	300	1	2.0291	.1065
2	3	1.0	.9	200	2	2.4675	.0917
2	3	1.0	.9	200	3	3.0256	.0873
2	3	1.414	.9	300	0	.9584	.1230
2	3	1.414	.9	300	1	1.3793	.0827
2	3	1.414	.9	200	2	2.1820	.0716
2	3	1.414	.9	200	3	2.9274	.0579
2	3	.707	.95	300	0	2.5170	.1063
2	3	.707	.95	300	1	2.6757	.1063
2	3	.707	.95	200	2	2.7982	.0955
2	3	.707	.95	200	3	3.0642	.0898
2	3	1.0	.95	300	0	1.8616	.1321
2	3	1.0	.95	300	1	1.9619	.0987
2	3	1.0	.95	200	2	2.3486	.0753
2	3	1.0	.95	200	3	2.9368	.0780
2	3	1.414	.95	300	0	.7027	.1042
2	3	1.414	.95	300	1	1.3557	.0868
2	3	1.414	.95	200	2	2.1514	.0580
2	3	1.414	.95	200	3	2.9337	.0341

TABLE A-4-4 (Continued)

c. Robustness

n	m	Dem.[1] S/N	Act.[2] S/N	Dem. ρ_{12}	Act. ρ_{12}	N	A	B	Nor. $\hat{MAE}$	$2\hat{\sigma}$ Nor. $\hat{MAE}$	Nor. $\hat{RMSE}$	$2\hat{\sigma}$ Nor. $\hat{RMSE}$
2	3	1.0	.707	.95	.95	700			.11218	.00942	.16769	.01314
2	3	1.0	.841	.95	.95	1200			.10255	.00689	.15732	.01046
2	3	1.0	1.19	.95	.95	2400			.059640	.003571	.10587	.00671
2	3	1.0	1.414	.95	.95	3000			.045072	.002813	.089255	.005971
2	3	1.0	1.0	.95	.9	1200			.082441	.006204	.13544	.01000
2	3	1.0	1.0	.95	.98	1600			.073740	.005084	.12561	.00864
2	3	2.0	1.414	.95	.95	4000			.030795	.002433	.082881	.006229
2	3	2.0	1.68	.95	.95	3400	1.25	.2	.016647	.001218	.053861	.003319
2	3	2.0	2.37	.95	.95	10000	1.25	.3	.0031472	.0002836	.019155	.001499
2	3	2.0	2.83	.95	.95	12000	1.4	.3	.0010672	.0001653	.010112	.001306
2	3	2.0	2.0	.95	.9	6000	1.25	.3	.012209	.001102	.046018	.003257
2	3	2.0	2.0	.95	.98	6000	1.25	.3	.0059579	.0004274	.028973	.001699

[1] Demodulator value.
[2] Actual value.

TABLE A-4-5

Data, Smith's Suboptimal Demodulator

n	m	S/N	ρ_{12}	N	Nor. $\hat{\text{MAE}}$	$2\hat{\sigma}$ Nor. $\hat{\text{MAE}}$	Nor. $\hat{\text{RMSE}}$	$2\hat{\sigma}$ Nor. $\hat{\text{RMSE}}$
2	3	.707	.9	400	.11871	.01346	.17947	.01725
2	3	1.0	.9	800	.083787	.008309	.14431	.01208
2	3	1.414	.9	1600	.038046	.004312	.094252	.01070
2	3	2.0	.9	4000	.010996	.001544	.050035	.007292
2	3	.707	.95	400	.11547	.01332	.17626	.01683
2	3	1.0	.95	800	.089814	.009548	.16217	.01318
2	3	1.414	.95	1600	.042898	.005247	.11336	.01161
2	3	2.0	.95	4000	.011548	.001766	.057026	.007859

TABLE A-4-6

Data, Rauch's Suboptimal Demodulator

a. Optimal Weighting Factors, m = 3 ($w_{4r} = w_{2r}$, $w_{5r} = w_{1r}$).

S/N	ρ_{12}	ρ_{13}	w_{11}	w_{21}	w_{31}	w_{12}	w_{22}	w_{32}	w_{13}	w_{23}	w_{33}
.707	.95	.85	.15	.19	.32	.15	.17	.36	.09	.15	.52
1.0	.95	.85	.12	.18	.40	.11	.16	.46	.08	.13	.58
1.414	.95	.85	.09	.14	.52	.08	.13	.58	.04	.11	.70
2.0	.95	.85	.05	.11	.68	.04	.10	.72	.03	.06	.82
2.83	.95	.85	.04	.06	.80	.03	.06	.82	.01	.04	.90
.707	.98	.95	.17	.20	.26	.16	.19	.30	.11	.19	.40
1.0	.98	.95	.15	.18	.34	.14	.17	.38	.09	.16	.50
1.414	.98	.95	.12	.15	.46	.10	.15	.50	.07	.13	.60
2.0	.98	.95	.07	.12	.62	.06	.12	.64	.04	.09	.74
2.83	.98	.95	.04	.08	.76	.04	.07	.78	.03	.05	.84
.707	.995	.99	.18	.21	.22	.17	.20	.26	.15	.19	.32
1.0	.995	.99	.17	.20	.26	.17	.18	.30	.14	.17	.38
1.414	.995	.99	.14	.16	.40	.13	.16	.42	.10	.15	.50
2.0	.995	.99	.09	.13	.56	.09	.12	.58	.06	.12	.64
2.83	.995	.99	.06	.08	.72	.06	.08	.72	.04	.07	.78

b. Performance, m = 3.

n	m	S/N	ρ_{12}	ρ_{13}	Nor. MAE	Nor. RMSE
5	3	.707	.95	.85	.12610	.22851
5	3	1.0	.95	.85	.076975	.17501
5	3	1.414	.95	.85	.036931	.11972
5	3	2.0	.95	.85	.010691	.064111
5	3	2.83	.95	.85	.0010973	.020520
5	3	.707	.98	.95	.10258	.20381
5	3	1.0	.98	.95	.055816	.14585
5	3	1.414	.98	.95	.024630	.095001
5	3	2.0	.98	.95	.0068462	.049710
5	3	2.83	.98	.95	.00069637	.015845
5	3	.707	.995	.99	.080245	.18122
5	3	1.0	.995	.99	.036012	.11711
5	3	1.414	.995	.99	.013487	.069558
5	3	2.0	.995	.99	.0034750	.034883
5	3	2.83	.995	.99	.00034044	.010880

TABLE A-4-6 (Continued)

c. Bit-error Probabilities (n = 5, m = 3) and Assumed Bit-error Probabilities (n = 5, m = 6).

S/N	ρ_{12}	ρ_{13}	PBE_1	PBE_2	PBE_3	Assumed PBE_4	Assumed PBE_5	Assumed PBE_6
.707	.95	.85	.12438	.14716	.20760	.21833	.22907	.23980
1.0	.95	.85	.068745	.085563	.13317	.14168	.15019	.15870
1.414	.95	.85	.031005	.039290	.064682	.069355	.074027	.07870
2.0	.95	.85	.0087543	.011022	.018484	.019889	.021295	.02270
2.83	.95	.85	.00089396	.0011146	.0018836	.0020224	.0021612	.00230
.707	.98	.95	.095509	.11910	.17283	.19515	.21748	.23980
1.0	.98	.95	.045061	.063087	.10538	.12315	.14093	.15870
1.414	.98	.95	.018179	.027061	.049520	.059247	.068973	.07870
2.0	.98	.95	.0048821	.0073722	.013948	.016865	.019783	.02270
2.83	.98	.95	.00049595	.00073893	.0014160	.0017107	.0020053	.00230
.707	.995	.99	.075698	.090042	.12202	.16128	.20054	.23980
1.0	.995	.99	.028846	.040025	.065105	.096303	.12750	.15870
1.414	.995	.99	.0094620	.014894	.027948	.044866	.061783	.07870
2.0	.995	.99	.0023165	.0037699	.0075961	.012631	.017665	.02270
2.83	.995	.99	.00022380	.00036580	.00075707	.0012714	.0017857	.00230

d. Performance, m = 6 (with assumed PBE_i, i = 4, 5, 6).

n	m	S/N	ρ_{12}	ρ_{13}	Nor. MAE	Nor. RMSE
5	6	.707	.95	.85	.12362	.20420
5	6	1.0	.95	.85	.078655	.15692
5	6	1.414	.95	.85	.039150	.10758
5	6	2.0	.95	.85	.011590	.057662
5	6	2.83	.95	.85	.0011978	.018459

TABLE A-4-6 (Continued)

d. Performance (Continued)

n	m	S/N	ρ_{12}	ρ_{13}	Nor. MAE	Nor. RMSE
5	6	.707	.98	.95	.10349	.18262
5	6	1.0	.98	.95	.059962	.13131
5	6	1.414	.98	.95	.027927	.085927
5	6	2.0	.98	.95	.0080005	.045034
5	6	2.83	.98	.95	.00082016	.014358
5	6	.707	.995	.99	.083861	.16270
5	6	1.0	.995	.99	.041876	.10613
5	6	1.414	.995	.99	.017380	.063614
5	6	2.0	.995	.99	.0047312	.032047
5	6	2.83	.995	.99	.00047219	.010010

e. Robustness

n	m	Dem. S/N	Act. S/N	Dem. ρ_{12}	Dem. ρ_{13}	Act. ρ_{12}	Act. ρ_{13}	Nor. MAE	Nor. RMSE
5	3	1.0	.707	.98	.95	.98	.95	.10534	.20689
5	3	1.0	1.414	.98	.95	.98	.95	.028227	.10232
5	3	1.0	2.0	.98	.95	.98	.95	.016025	.078231
5	3	1.0	1.0	.98	.95	.95	.85	.078972	.17719
5	3	1.0	1.0	.98	.95	.995	.99	.037956	.11984
5	3	1.414	.707	.98	.95	.98	.95	.11846	.22168
5	3	1.414	1.0	.98	.95	.98	.95	.061554	.15471
5	3	1.414	2.0	.98	.95	.98	.95	.0096905	.060197
5	3	1.414	2.83	.98	.95	.98	.95	.0045584	.042902
5	3	1.414	1.414	.98	.95	.95	.85	.038998	.12322
5	3	1.414	1.414	.98	.95	.995	.99	.014498	.07208
5	3	2.0	1.0	.98	.95	.98	.95	.080085	.18107

TABLE A-4-6 (Continued).

e. Stability (Continued).

n	m	Dem. S/N	Act. S/N	Dem. ρ_{12}	Dem. ρ_{13}	Act. ρ_{12}	Act. ρ_{13}	Nor. MAE	Nor. RMSE
5	3	2.0	1.414	.98	.95	.98	.95	.029933	.10714
5	3	2.0	2.83	.98	.95	.98	.95	.0012841	.022316
5	3	2.0	2.0	.98	.95	.95	.85	.011399	.066346
5	3	2.0	2.0	.98	.95	.995	.99	.0038908	.036963

TABLE A-4-7

Data, Suboptimal Demodulator No. 3

n	m	S/N	ρ_{12}	ρ_{13}	WT	N	Nor. $\hat{MAE}$	$2\hat{\sigma}$ Nor. $\hat{MAE}$	Nor. $\hat{RMSE}$	$2\hat{\sigma}$ Nor. $\hat{RMSE}$
5	3	.707	.95	.85	3.5	600	.097991	.008971	.14722	.01247
5	3	1.0	.95	.85	2.5	1000	.063546	.005623	.10929	.008757
5	3	1.414	.95	.85	1.5	1600	.039379	.003252	.076029	.005492
5	3	2.0	.95	.85	1.5	3500	.011157	.001336	.041076	.004743
5	3	.707	.98	.95	4.0	700	.077441	.007453	.12537	.01117
5	3	1.0	.98	.95	3.0	1100	.047136	.004638	.090205	.008248
5	3	1.414	.98	.95	2.0	2000	.026955	.002703	.066180	.006172
5	3	2.0	.98	.95	1.5	5500	.0092144	.0009483	.036352	.003491
5	3	.707	.995	.99	4.5	900	.070494	.006612	.12168	.009932
5	3	1.0	.995	.99	3.5	2200	.033276	.002995	.077716	.006340
5	3	1.414	.995	.99	2.5	5000	.015423	.001356	.050374	.003906
5	3	2.0	.995	.99	1.5	10000	.0055604	.0005430	.027714	.002200

TABLE A-4-8

Computation Times

Demodulator	m	n	Execution Time for 100 Iterations	Total Execution Time
Present Day	3/6	1		5.0 sec.
Optimal	3	1	1.66 sec.	1 min. 22.3 sec.
Optimal	3	2	3.0 sec.	34 min. 1.8 sec.[1]
Optimal	3	3	11.8 sec.	22 min. 14.2 sec.
Optimal	6	1	11.7 sec.	7 min. 58.2 sec.
Optimal	6	2	41.9 sec.	130 min. 22.6 sec.[1]
Smith's	3	2	5.0 sec.	11 min. 20.3 sec.
No. 3	3	5	3.2 sec.	17 min. 57.4 sec.
Rauch's	3	5		21 min. 8.3 sec.[2]
			Total:	4 hr. 6 min. 30.1 sec.

[1] Includes bias and robustness runs.

[2] Includes bias and robustness runs and run for m = 6 with assumed PBE_r's.

Bibliography

Reference
Designation

B1 Blackwell, David and Girshick, M.A., Theory of Games and Statistical Decisions, New York; John Wiley and Sons, Inc. 1954.

C1 Cramer, Harold, The Elements of Probability Theory, New York; John Wiley and Sons, Inc., 1955.

D1 Davenport, W. B. Jr., and Root, W.L., An Introduction to the Theory of Random Signals and Noise, New York; McGraw-Hill Book Company, Inc., 1958.

K1 Kahn, Herman, "Use of Different Monte Carlo Sampling Techniques." Symposium on Monte Carlo Methods; New York; John Wiley and Sons, Inc., 1956, pp. 146-190.

K2 Kahn, Herman, Applications of Monte Carlo, RM-1237-AEC, The Rand Corp., 19 Apr 1954, revised 27 Apr 1956.

M1 McCormick, J.M., and Salvadori, M.G., Numerical Methods in FORTRAN, Englewood Cliffs, New Jersey; Prentice-Hall, Inc., 1965.

M2 McCracken, D.D., A Guide to FORTRAN Programming, New York; John Wiley and Sons, Inc., 1961.

M3 McRae, D.D., Interpolation Errors. Technical Report 1, Parts 1 and 2, Advanced Telemetry Study, Radiation Incorporated, Melbourne, Florida, Part 1: 15 February 1961, Part 2: 16 March 1961.

M4 McRae, D.D., and Smith, E. F., Computer Interpolation. Technical Report No. 2, Parts 1 and 2, Advanced Communications Group, Radiation Incorporated, Melbourne, Florida, Part 1: 15 December 1961, Part 2: 10 May 1962.

M5 Mood, A.M., Introduction to the Theory of Statistics, New York; McGraw-Hill Book Co., Inc., 1950.

S1 Smith, Earl F., "Minimum-Error Probabilities in Demodulation of Binary PCM Signals." IEEE Transactions on Space Electronics and Telemetry, vol. SET-10, no. 4 (December 1964), pp. 135-143.

S2 Smith, Earl F., Minimum-Error Demodulation of Binary PCM Signals, Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan, 1963.

U1 University of Michigan, University of Michigan Executive System for the IBM 7090 Computer. 3 Vols. September 1965.

W1 Woodward, P.M., Probability and Information Theory with Applications to Radar, New York; McGraw-Hill Book Co., Inc., 1953.

List of Tables

List of Appendices

List of Figures